

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Prototype de système expert pour la planification d'expériences en biochimie

Demeyer, C.; Trinteler, Marie-France

Award date:
1989

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

**Facultés Universitaires Notre-Dame de la Paix
Institut d'Informatique**

**PROTOTYPE DE SYSTEME EXPERT
POUR LA PLANIFICATION
D'EXPERIENCES EN BIOCHIMIE**

C.DEMEYER et M-F.TRINTELER

**Mémoire présenté en vue de l'obtention
du titre de licencié et maître en
Informatique**

**Promoteur : M.NOIRHOMME-FRAITURE
Directeur : J.BARRETO**

Septembre 1989

Facultés Universitaires Notre-Dame de la Paix
Institut d'Informatique
rue de Bruxelles, 61, B-5000 NAMUR

PROTOTYPE DE SYSTEME EXPERT POUR LA PLANIFICATION
D'EXPERIENCES EN BIOCHIMIE

C.DEMEYER et M-F.TRINTELER

Résumé

Ce travail a pour objet la conception d'un prototype de système expert pour la planification d'expériences en biologie. Il s'inscrit dans le cadre d'une collaboration avec le laboratoire de biochimie cellulaire des Facultés, collaboration qui a déjà donné lieu à une étude d'opportunité, laquelle nous a servi de point de départ. Lors de l'implémentation, nous avons considéré six techniques de séparation de protéines - tamis moléculaire, chromatographie sur échangeur d'ions, précipitation isoélectrique, électrophorèse, électrofocalisation et HPLC - que nous avons intégrées dans le but de pouvoir élaborer un plan de purification. Conjointement à ce travail, nous avons réalisé une analyse comparative de trois outils de développement de logiciels : VP-Expert - shell avec lequel nous avons construit notre programme -, Nexpert et Guru.

Abstract

The goal of this study is to develop a prototype of expert system for the planification of biochemical experiments. It takes place in the frame of the collaboration with cellular biochemistry laboratory of the University. This collaboration led to a feasibility analysis and serves as starting point. During the implementation, six techniques of proteins separation have been viewed. These are the gel filtration, the ion exchange chromatography, the isoelectric precipitation, the electrophoresis, the electrofocusing and the HPLC. Simultaneously with this work, we achieved a comparative analysis of three shells, namely VP-Expert - which was used to build our program -, Nexpert and Guru.

Mémoire de licence et maîtrise en Informatique
Septembre 1989
Promoteur : M.NOIRHOMME-FRAITURE
Directeur : J.BARRETO

Nos remerciements s'adressent tout naturellement à notre promoteur, Madame M.Noirhomme, ainsi qu'au directeur de notre mémoire, Monsieur J.Barreto, qui ont accepté de superviser notre travail.

Nous tenons à remercier tout particulièrement le Professeur J.Remacle du laboratoire de biochimie cellulaire à la Faculté des Sciences pour ses conseils judicieux et sa constante disponibilité qui nous ont permis de progresser tout au long de cette année.

Nous remercions également tous les autres membres du laboratoire de biochimie pour l'accueil qu'ils nous ont réservé, et plus spécialement Monsieur E.Delaive pour sa patience et ses conseils techniques.

Nous souhaitons également exprimer notre reconnaissance à l'égard de Mesdemoiselles K.Becker, M.Chandelon, et R.Lessa Pagano pour leur aide précieuse apportée lors de la rédaction de ce mémoire.

Nous exprimons toute notre gratitude à nos amies et amis qui nous ont épaulées durant ce travail.

Enfin, nous exprimons notre reconnaissance à nos parents qui, plus discrètement, nous ont secondées au cours de nos études, et nous ont apporté tant l'aide matérielle que morale et affective, nécessaire à la réalisation de ce travail.

TABLE DES MATIERES

<u>Introduction</u>	1
<u>I. Contexte biologique</u>	3
1. Origine de la collaboration	3
2. Difficultés et problèmes	4
3. Techniques de purification des protéines	6
3.1. Concepts théoriques	7
3.1.1. Le pH	7
3.1.2. Le tampon	7
3.1.3. La technique de chromatographie	8
3.1.4. L'électrophorèse	10
3.2. Techniques de purification	12
3.2.1. Procédés de séparation basés sur la taille moléculaire	12
a. Chromatographie sur tamis moléculaire	13
b. HPLC	19
c. Dialyse et ultrafiltration	20
d. Centrifugation en gradient de densité	20
3.2.2. Procédés de séparation basés sur la solubilité ..	21
a. Précipitation isoélectrique	21
b. Dissolution par les sels et relargage des protéines	24
c. Fractionnement par des solvants	25
d. Effet de la température sur la solubilité des protéines	25

3.2.3. Procédés de séparation basés sur la charge électrique	25
a. Chromatographie sur échangeur d'ions	26
b. Les méthodes électrophorétiques	29
3.2.4. Autres types de chromatographie	35
a. Chromatographie sur papier	35
b. Chromatographie d'affinité	36
c. Chromatographie sur couche mince	36
d. Chromatographie en phase gazeuse	37
e. Conclusion	38

II. Contexte informatique39

1. Introduction à l'intelligence artificielle	39
1.1. Définition et domaines de l'intelligence artificielle	39
1.1.1. Le langage naturel	40
1.1.2. La robotique avancée	40
1.1.3. L'amélioration des interfaces homme-machine	41
1.1.4. La programmation explorative	41
1.2. Les idées clés de l'intelligence artificielle	42
1.2.1. La représentation des connaissances	42
1.2.2. La recherche heuristique	43
1.2.3. Séparation des connaissances d'inférence et de contrôle	44
1.3. Conclusion	44
2. Introduction aux systèmes experts	46
2.1. Approche des systèmes experts	46
2.2. Définition d'un système expert et composants	47
2.3. La connaissance et le raisonnement d'un système expert	49
2.4. L'explication du raisonnement	50

2.5. Types de systèmes experts	50
2.5.1. L'interprétation	50
2.5.2. La prédiction	51
2.5.3. Le diagnostic	51
2.5.4. La planification	51
2.5.5. Le monitoring	51
2.5.6. Le contrôle	52
2.5.7. L'instruction	52
2.5.8. La mise au point	53
2.6. Choix du domaine d'application d'un système expert ..	53
2.6.1. Caractéristiques de base	54
2.6.2. Type de problème	54
2.6.3. L'expert	55
2.6.4. Autres caractéristiques souhaitables	55
2.7. Une méthode de construction d'un système expert	56
2.7.1. L'étape d'identification	57
a. Identification des participants et de leur rôle ...	57
b. Identification du problème	58
c. Identification des ressources	59
d. Identification des objectifs	60
2.7.2. L'étape de conceptualisation	60
2.7.3. L'étape de formalisation	60
2.7.4. L'étape d'implémentation	61
2.7.5. L'étape de test	61
2.7.6. La mise au point du prototype	62
2.8. Evaluation d'un système expert	63
2.8.1. Vérification, validation et évaluation	64
2.8.2. La validation des systèmes experts	65
a. Valider quoi ?	65
b. Valider contre quoi ?	65
c. Valider avec quoi ?	66
d. Quand valider ?	66
e. Comment contrôler le coût de la validation ?	67
f. Comment traiter les résultats multiples ?	67
2.8.3. Quelques concepts fondamentaux de la validation	67

2.8.4. La validation qualitative	68
a. Validation de "surface"	68
b. Tests de Turing	69
c. Tests dans le domaine d'application	69
d. Validation des sous-systèmes	69
e. Interaction visuelle	69
2.9. Conclusions	71
3. Langages et outils de construction de systèmes experts	73
3.1. Langages	73
3.1.1. Les paradigmes de programmation	74
a. Style déclaratif versus style impératif	75
b. La programmation logique	76
c. La programmation fonctionnelle	76
d. La programmation orientée objet	77
3.1.2. Software symbolique et software conventionnel ...	77
3.1.3. Programmation conventionnelle et ingénierie de connaissances	79
3.1.4. Lisp	82
3.1.5. Prolog	84
3.1.6. Langages orientés objet	87
3.2. Outils de développement	88
3.2.1. Classification des outils	88
a. Outils d'induction	89
b. Outils à base de règles simples ou structurées	90
c. Outils hybrides	92
d. Outils dédiacés	93
3.2.2. Outils à base de règles versus outils orientés objet	93
3.3. Analyse de trois outils : VP-Expert, Nexpert et Guru	94
3.3.1. Critères d'analyse	95
a. Représentation des connaissances, inférence et contrôle	95
b. Interface développeur	96
c. Interface utilisateur	97
d. Interface système	97

e. Software	97
f. Hardware	98
g. Documentation	98
3.3.2. Analyse comparative	98
a. VP-Expert	98
b. Nexpert	105
c. Guru	113
3.3.3. Synthèse	124
3.4. Marché, applications et exemples de systèmes experts	126
3.4.1. MYCIN	128
3.4.2. XCON	129
3.4.3. DENDRAL	129
 III. <u>Présentation du travail</u>	131
1. Aperçu du travail réalisé antérieurement	131
2. Méthode d'investigation	133
2.1. Identification du problème et des acteurs	133
2.1.1. Identification du problème et de l'environnement de travail	133
2.1.2. Identification des acteurs	137
2.1.3. Spécificités d'un système expert de planification	138
2.1.4. Objectifs du système	140
2.2. Conceptualisation	140
2.2.1. Concepts et relations	140
2.2.2. Tâches	142
2.2.3. Critères pour déterminer l'application d'une technique	145
a. La précipitation	145
b. La chromatographie sur échangeur d'ions	147
c. La chromatographie sur tamis moléculaire	155
d. L'HPLC	166
e. L'électrofocalisation	167
f. L'électrophorèse	168

2.2.4. Les enchainements des techniques de purification	169
a. Un exemple d'enchainement logique des méthodes de séparation	169
b. Critères d'enchainement des techniques de purification	171
c. Perspectives de développement	172
2.3. Implémentation	173
2.3.1. Moteur d'inférence	173
2.3.2. Stockage des données	175
2.3.3. Composition du programme	179
 IV. Conclusions	184
1. Critiques de l'outil	184
2. Critiques du travail	186
3. Perspectives d'avenir	187
 - Glossaires	189
- Bibliographie	201
- Annexe : code du programme	

Remarque : il est possible de disposer d'un manuel d'utilisation.

INTRODUCTION

L'homme a souvent pensé que les machines ne pourraient pas raisonner.

Cependant, face aux progrès technologiques de ces quinze dernières années, il est amené à revoir sa position.

En effet, depuis plus d'une décennie, les chercheurs essaient de repousser les limites des ordinateurs. Pour cela, ils ont créé des machines capables, non seulement de stocker l'information mais également d'en retirer de la valeur ajoutée.

C'était l'aube de l'intelligence artificielle, le début des machines qui peuvent exécuter des processus de raisonnement semblables à ceux de l'homme.

Actuellement, l'intelligence artificielle inclut différents domaines comme le langage naturel, la robotique, la reconnaissance des formes, mais c'est le secteur des systèmes experts - logiciels qui emmagasinent des connaissances d'experts humains, et qui sont capables de résoudre des problèmes spécifiques - qui a suscité notre intérêt.

L'objectif de notre travail a consisté en la réalisation d'un prototype de système expert aidant à la planification d'expériences en biochimie, plus particulièrement dans le domaine de la purification de substances biologiques. Ce programme a été écrit au moyen d'un outil de développement de systèmes experts appelé VP-Expert.

Notre travail est découpé en trois parties.

La première partie est consacrée à l'étude bibliographique des concepts et notions rencontrés dans les laboratoires de biochimie. Nous décrirons les fondements biologiques de l'application en mettant l'accent sur les techniques que nous avons implémentées.

La deuxième partie décrit le cadre conceptuel, à savoir l'intelligence artificielle, dans lequel ce travail s'inscrit. Après avoir introduit les principes de cette nouvelle discipline, nous ferons une synthèse des langages et outils que l'informatique met à la disposition du concepteur de systèmes experts.

Dans une troisième partie, nous décrirons la méthode d'investigation que nous avons suivie tout au long de cette année. Celle-ci s'est déroulée en harmonie avec le laboratoire de biochimie cellulaire de la Faculté des Sciences, et nous a permis d'aboutir à un système de connaissances formalisées, lesquelles, nous l'espérons, serviront de base pour une exploitation ultérieure.

En guise de conclusion, nous nous proposons de jeter un regard critique sur ce qui a été réalisé et sur les améliorations susceptibles de constituer une suite à ce travail.

Enfin, deux glossaires, l'un biologique l'autre informatique, permettront aux non-initiés dans l'une ou l'autre de ces disciplines de pénétrer l'univers qui leur est étranger.

Le lecteur trouvera en annexe le code du programme, ainsi qu'un manuel d'utilisation du logiciel.

I. CONTEXTE BIOLOGIQUE

I. CONTEXTE BIOLOGIQUE

Ce chapitre a pour objet de présenter d'une manière synthétique le domaine dans lequel nous avons travaillé, ceci afin de faciliter la compréhension du lecteur non familiarisé aux concepts biochimiques.

Dans cette première partie, après avoir brièvement traité du cheminement qui a permis d'arriver à une collaboration avec le laboratoire de biochimie, nous exposerons quelques difficultés rencontrées lors de la manipulation de matériel biologique, et ensuite nous développerons les techniques de purification de protéines sur lesquelles s'est basé notre travail.

1. Origine de la collaboration

L'année dernière, deux étudiants en informatique s'étaient déjà familiarisés avec les activités des biochimistes du laboratoire du Professeur J.Remacle à la Faculté des Sciences dans le but de développer un système d'aide à la planification d'expériences.

Une telle collaboration peut évidemment présenter des intérêts non seulement pour les informaticiens, mais surtout pour les biochimistes pour qui l'outil informatique est susceptible de faciliter les interprétations de résultats d'expériences, l'organisation et la coordination de différentes

tâches ou étapes d'un travail complexe, le stockage d'informations diverses,...

De plus, le travail en laboratoire n'est actuellement possible que grâce à l'utilisation d'une technologie de pointe faisant souvent appel à l'informatique.

Un an pour effectuer un tel système d'aide, c'est bien sûr trop court. C'est pourquoi le sujet a été proposé à nouveau afin d'en étendre l'implémentation.

Il faut noter que nous partions avec un certain avantage par rapport aux deux autres étudiants : la connaissance du domaine biochimique, du moins en ce qui concerne les bases techniques ! Les différentes phases du travail que nous avons réalisé seront développées au cours du troisième chapitre.

Analysons dès maintenant les difficultés qui apparaissent lors du travail de purification, et les problèmes liés aux techniques utilisées et à leur choix.

2. Difficultés et problèmes

Rappelons-nous que les protéines sont des macromolécules de poids moléculaire bien défini, qui peuvent donc former des solutions moléculaires vraies et qui sont des électrolytes dont le comportement peut se comprendre à partir des principes physiques qui régissent les électrolytes de petite taille.

Grâce aux efforts de nombreux chercheurs, l'étude du comportement des protéines en solution et de leur séparation est devenue quasiment une science exacte.

Ceci est surtout intéressant en biochimie, car dans ce domaine une part importante du travail nécessite l'isolement de composants d'origine biologique. Le succès des recherches dépend donc d'une préparation judicieuse des substances cellulaires et du choix de méthodes de séparation adéquates.

La difficulté du travail de purification réside dans la mise au point et l'utilisation de techniques raffinées. En effet, un mélange biochimique est composé d'un grand nombre de substances ayant des propriétés chimiques assez similaires et qui sont sensibles à beaucoup de facteurs physico-chimiques comme la température, le pH, etc.

L'isolement d'un composant biologique à partir d'un mélange peut se faire suivant différents critères comme nous allons le détailler ci-dessous et il est d'autant meilleur que les valeurs obtenues pour les différents éléments du mélange concernant une même propriété sont éloignées les unes des autres.

Les techniques de purification sont utiles au niveau de la recherche et de l'industrie, tant pour des travaux préparatifs qu'analytiques. Dans les travaux de préparation, il s'agit de purifier des substances afin de les utiliser par la suite, alors que les travaux analytiques ont pour objectif l'identification et la caractérisation des diverses substances d'un mélange.

Dans la littérature, ces techniques sont présentées comme des "recettes" où sont rarement exposés les raisonnements qui ont accompagné leur mise au point. Ceci peut s'expliquer par le fait que les composants d'un mélange dont on souhaite extraire une substance sont parfois mal connus et que donc des réactions peuvent échapper au chercheur. De ce fait, celui-ci peut très bien constater par essais et erreurs l'efficacité d'une purification sans pouvoir l'expliquer dans l'immédiat. De plus, la "recette" présentée n'est souvent qu'une base pour le scientifique ; il doit ajuster lui-même, avec précision, la purification aux conditions de travail qui sont les siennes et ceci demande du temps et du savoir-faire.

Souvent, le chercheur fera appel à des personnes expérimentées dans les diverses techniques et qui peuvent apporter une expérience pratique indispensable ; malheureusement, de telles "sources" ne sont pas toujours présentes dans un laboratoire et, si elles existent, elles sont parfois indisponibles.

En réalité, les connaissances sont distribuées chez un grand nombre de personnes, chacune ayant une expérience plus ou moins importante dans la manipulation de certaines techniques, et de certains mélanges.

Après avoir précisé ces quelques points, nous allons étudier différentes techniques de purification de protéines.

3. Techniques de purification des protéines

Bien que l'on connaisse les protéines depuis plus de cent ans, nos notions actuelles sur leur comportement comme solutés dans les systèmes aqueux n'ont été acquises qu'au cours de ces dernières années par l'étude intensive de leur physico-chimie.

Dans les sections qui suivent, nous introduirons brièvement quelques concepts théoriques, ensuite nous étudierons les techniques de purification et nous verrons comment les différentes propriétés caractéristiques des protéines globulaires en solution peuvent être exploitées pour séparer un mélange de protéines selon : (1) leur taille moléculaire, (2) leur solubilité, (3) leur charge électrique, (4) leurs différences d'adsorption caractéristiques et (5) leur affinité biologique pour d'autres molécules.

3.1. Concepts théoriques

Pour faciliter la compréhension, nous avons jugé intéressant de définir quelques notions biochimiques de base. Précisons dès à présent qu'un glossaire plus complet a également été rédigé ; il précède les annexes de ce travail.

3.1.1. Le pH

Le pH d'une solution est une mesure de la concentration en ions H_3O^+ de cette solution :

$$pH = -\log [H_3O^+]$$

où $[x]$ représente la concentration de x .

Le pH permet donc de caractériser l'acidité ou la basicité. En effet, dans une solution acide, la concentration en ions H_3O^+ est supérieure à la concentration en ions OH^- et le pH est inférieur à 7.

Une solution basique présente au contraire un pH supérieur à 7 ($[H_3O^+] < [OH^-]$).

A 25°C par exemple, la concentration en ions H_3O^+ d'une solution neutre est égale à 10^{-7} molaire et le pH vaut 7.

3.1.2. Le tampon

Lorsqu'on manipule des substances biologiques comme c'est le cas lors d'une purification, il est important de stabiliser le pH du système, car ces substances sont très sensibles à des changements de pH. De telles variations peuvent dénaturer les protéines.

Il est dès lors nécessaire de disposer de milieux capables de conserver un pH déterminé et constant malgré l'addition d'ions H_3O^+ ou OH^- dans le système. Ces milieux sont appelés les tampons.

Un tampon dispose donc d'une capacité plus ou moins grande à résister à une variation importante de pH. Cette capacité est appelée le pouvoir tampon. A chaque tampon est associée une zone de pouvoir tampon ; il s'agit d'une zone de pH à l'intérieur de laquelle le tampon contrôle bien les fluctuations de pH. Elle est définie par l'intervalle $[pK-1, pK+1]$; le pK d'un tampon étant le pH duquel une forme acide (ou basique) est à moitié dissociée.

3.1.3. La technique de chromatographie

En chimie, et plus particulièrement en biochimie, il est souvent nécessaire de séparer les composants d'un mélange plus ou moins complexe. Il existe plusieurs techniques de séparation : la distillation, la cristallisation, l'extraction,...

Cependant, la chromatographie permet de séparer et de détecter assez rapidement des produits en solution qui sont présents en quantités infimes. De plus, le haut degré de pureté des constituants qui est obtenu rend cette technique très intéressante.

Le terme chromatographie signifie "écrire avec les couleurs", car à l'origine on utilisait cette méthode pour séparer les pigments de plantes ou de colorants. Aujourd'hui, la chromatographie est aussi utilisée pour isoler des composants incolores. Il existe toute une série de techniques de chromatographie. Prenons par exemple le cas le plus simple de la chromatographie liquide/solide qui nous permettra de comprendre le principe de cette technique : cette chromatographie est une

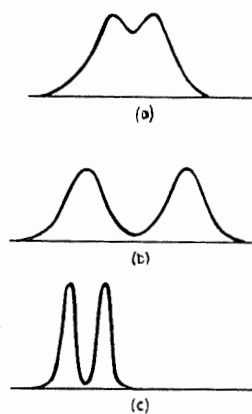


Figure 1 :

Résolution dans la chromatographie :

(a) faible résolution

(b) bonne résolution résultant d'une bonne sélectivité

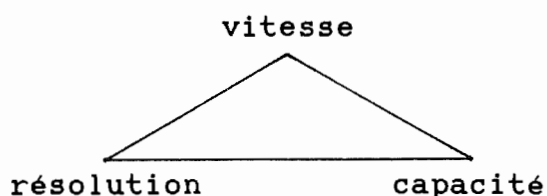
(c) bonne résolution résultant d'une bonne efficacité de la colonne

[Encyclopedia, 1987]

méthode de séparation par laquelle les substances d'un mélange sont emportées par un liquide à des vitesses différentes sur un support solide pouvant varier par la forme et la nature.

Le principe de la méthode est le suivant : après avoir déposé le mélange de substances sur la colonne ou support (ou encore adsorbant, phase stationnaire), on y fait passer un liquide (ou encore solvant, éluant, phase mobile ; cette phase est appelée élution). Les substances contenues dans le mélange initial sont alors entraînées à des vitesses différentes et sont retenues à des endroits qui varient en fonction des caractéristiques particulières propres à ces substances définissant le type de lien les unissant au substrat. Il se forme alors des zones pour chaque substance ; ces zones doivent être révélées par un processus physique ou chimique à déterminer.

Le résultat obtenu est appelé le chromatogramme. A ce niveau, la résolution est un facteur important et doit être maximale (voir figure 1). Elle dépend en fait d'autres paramètres comme la taille des particules de la phase stationnaire et/ou la vitesse de la phase mobile. Outre la résolution, la vitesse de passage et la capacité de la colonne (ou du support solide en général) - c'est-à-dire la taille maximale de l'échantillon à déposer - retiennent l'attention du scientifique. En effet, pour obtenir un résultat optimal, il est souhaitable que ces trois paramètres atteignent des valeurs les plus élevées possibles. Toutefois, il faut préciser que celles-ci ne peuvent pas être maximales simultanément ; un compromis s'impose donc pour résoudre ce dilemme "triangulaire" :



La chromatographie préparative permet l'identification des constituants et leur dosage. En outre, les séparations par cette technique peuvent, et souvent doivent, être effectuées sur de très petites quantités de substances, ce qui constitue un avantage appréciable.

Rappelons aussi deux grands principes en chromatographie :

- la chromatographie par adsorption est basée sur la différence d'adsorption des constituants d'une phase mobile sur une phase fixe qu'ils traversent, alors que
- la chromatographie de partage repose sur le coefficient de partage d'un corps dans deux solvants non miscibles.

Ces différentes techniques seront étudiées plus en détails dans la section suivante.

3.1.4. L'électrophorèse

Cette technique est basée sur la migration de particules chargées en solution dans un champ électrique. En effet, beaucoup de solutions aqueuses, surtout les acides, les bases et les sels, sont capables de "transporter" de l'électricité. Ce sont des électrolytes.

Quand un électrolyte est dissout dans l'eau, ses molécules se dissocient en fragments chargés (les ions). La conduction est rendue possible quand des électrodes sont placées dans la solution et entraînent un mouvement ionique.

Nous allons maintenant relever quelques facteurs critiques relatifs à cette technique.

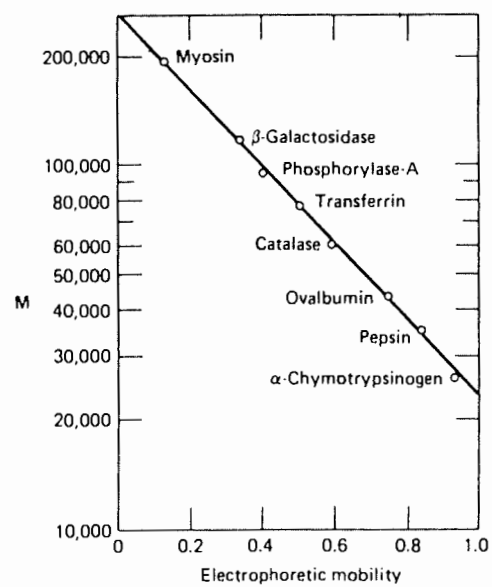


Figure 2 :
Corrélation entre le poids moléculaire (échelle logarithmique)
et la mobilité électrophorétique des protéines.
[Encyclopedia, 1987]

Dans les conditions idéales (particules dont la dilution est infinie et dans un milieu isolant dépourvu de sels), la force d'intensité F qui s'exerce sur une particule de charge globale Q soumise à un champ électrique E est telle que

$$F = Q.E.$$

De ce fait, l'accélération de la particule est constante si E reste constant lui aussi.

Mais, en réalité, la viscosité du milieu introduit une résistance qui croît linéairement avec la vitesse de la particule. Cette résistance dépend également du poids moléculaire et de la géométrie de celle-ci.

En pratique, il faut tenir compte du fait que les particules ne sont pas idéalement sphériques et que l'électrophorèse a lieu dans une solution saline non isolante ; les ions du solvant, de signe opposé à celui de la particule viennent neutraliser partiellement les charges de la protéine. De plus elles se déplacent dans le sens contraire du mouvement de cette dernière et réduisent donc la mobilité. Cette conséquence peut être également observée lorsque le liquide qui entoure la particule migratrice se déplace dans le sens contraire de celui de la migration.

La capacité des différents électrolytes à exercer un effet retardateur ne dépend pas de la nature chimique des ions, mais bien de leur concentration et de leur charge, donc dans l'ensemble de la force ionique de la solution : la mobilité électrophorétique diminue lorsque cette force augmente. Ajoutons qu'elle varie également en fonction du PM de la particule (voir figure 2).

Pour que la concentration en ions hydrogène (exprimée par le pH) du milieu se maintienne pendant toute la durée de la migration, il est nécessaire de travailler avec une solution pourvue d'un pouvoir tampon suffisant, mais qui ne diminue pas trop la mobilité électrophorétique.

3.2. Techniques de purification

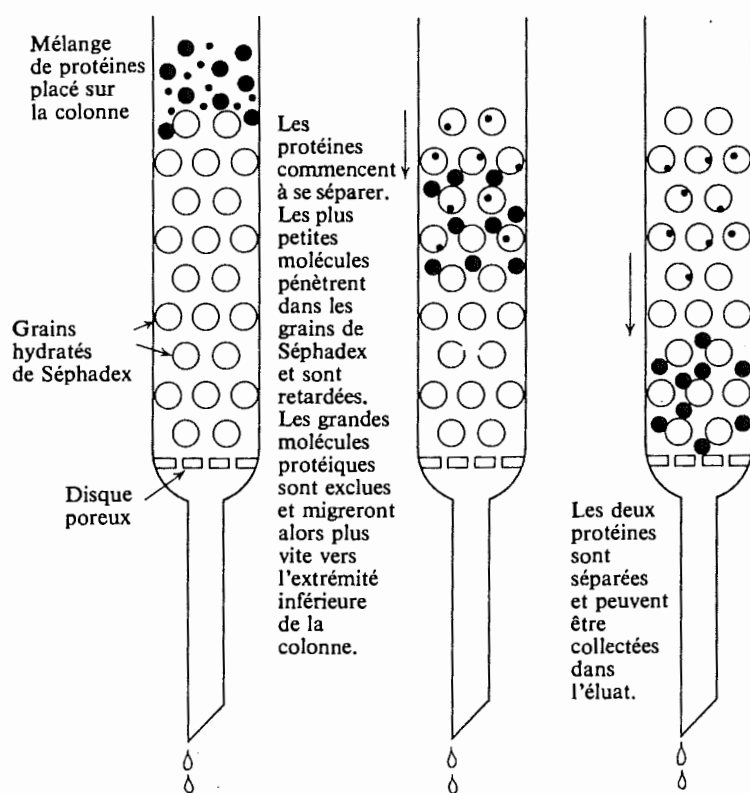
L'utilisation optimale d'une technique requiert une bonne compréhension des théories qui sont sous-jacentes. Cette connaissance permet au scientifique d'apprécier les capacités et les faiblesses de la technique et d'être critique lors de l'interprétation des résultats obtenus. C'est pourquoi nous allons reprendre les principes de base de quelques techniques.

Précisons dès à présent que seules cinq techniques retiendront notre attention lors de l'implémentation ; il s'agit de la précipitation isoélectrique, la chromatographie sur échangeurs d'ions, la chromatographie sur tamis moléculaire, l'HPLC et l'électrophorèse.

3.2.1. Procédés de séparation basés sur la taille moléculaire

La caractéristique la plus évidente des protéines est leur grande taille qui rend possible l'application de méthodes simples permettant d'isoler les protéines des petites molécules ou de séparer un mélange de protéines.

Séparation de deux protéines de taille différente sur une colonne de Séphadex.



Agrandissement montrant le processus d'exclusion

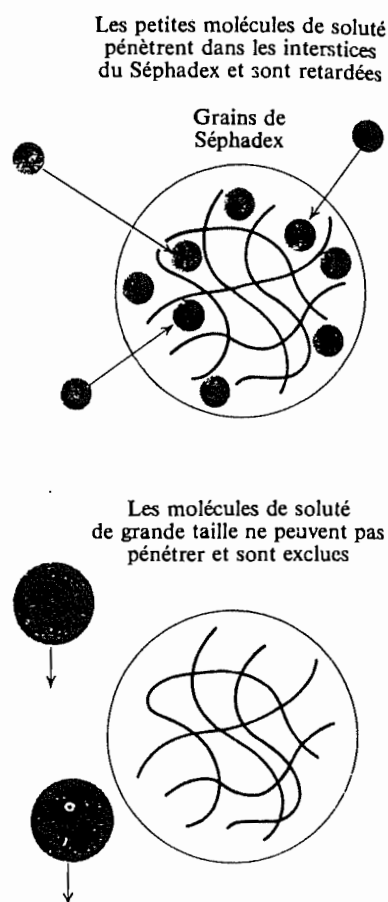


Figure 3 :
Principe du tamis moléculaire.
[Lehninger A.L., 1977]

a. La chromatographie sur tamis moléculaire (filtration sur gel)

a.1. Principe et théorie

La chromatographie sur tamis moléculaire est une méthode récente pour séparer des macromolécules sur base de leur différence de poids moléculaire. Le gel est une structure tridimensionnelle constituée d'un matériel solide se trouvant entouré d'un liquide. En utilisant des groupements réactionnels adéquats, on peut réaliser des gels avec presque toutes les macromolécules. Les gels sont formés de grains possédant des pores de dimensions variables.

Les molécules de taille supérieure aux plus gros pores ne peuvent pénétrer dans les grains du gel : elles vont descendre la colonne dans la phase liquide circulant autour des grains et vont être éluées dans la colonne. Ci-contre se trouve une illustration schématique de cette explication (voir figure 3).

Le résultat de ce type de chromatographie sera une séparation des molécules suivant l'ordre décroissant de leur taille.

Au point de vue théorique, la chromatographie sur tamis moléculaire peut se traiter comme une chromatographie ordinaire en considérant que la distribution entre la phase liquide et le support solide résulte non pas d'une adsorption sur le support solide mais d'une élimination de la phase liquide qui dépend de la taille de la molécule. On peut donc définir un coefficient de partage entre la phase liquide et la phase gel :

$$K_{av} = \frac{V_e - V_o}{V_t - V_o}$$

où V_e = volume d'élution du composé X

V_o = volume mort ou vide

V_t = volume total de la colonne.

Notons que pour rester cohérentes avec les notations employées dans le logiciel, nous avons décidé de ne pas représenter ces variables de manière indicée.

On remarquera que plus une substance sera grosse et donc exclue du gel, plus K_{av} sera petit ; et réciproquement, plus une substance pourra pénétrer le gel, plus K_{av} sera grand.

a.2. Différents types de gels

Les gels se distinguent par leur indice de rétention d'eau, qui caractérise leur aptitude à retenir l'eau, d'où la taille des molécules ralenties. La taille du grain intervient dans la vitesse de l'écoulement et dans la finesse des séparations. Les grains fins donnent des débits faibles et de bonnes résolutions. Les gros grains conviennent pour des séparations préparatives.

Les gels les plus fréquemment utilisés et les plus commercialisés sont :

1°) le Sephadex

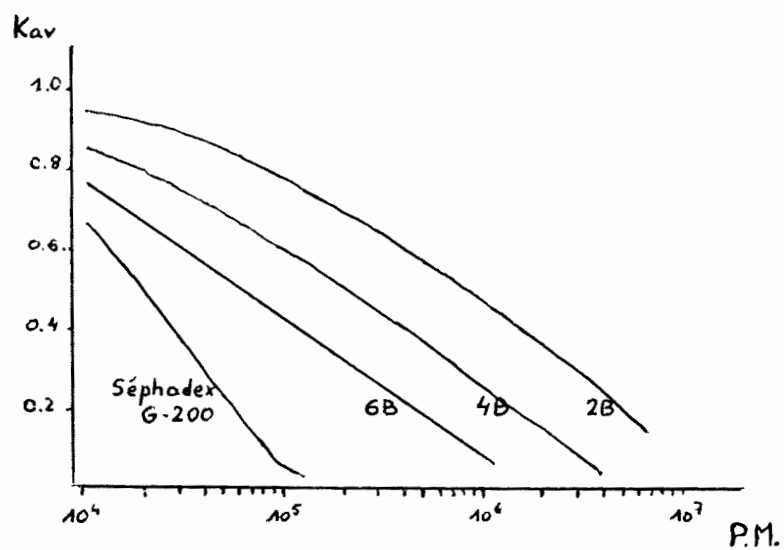
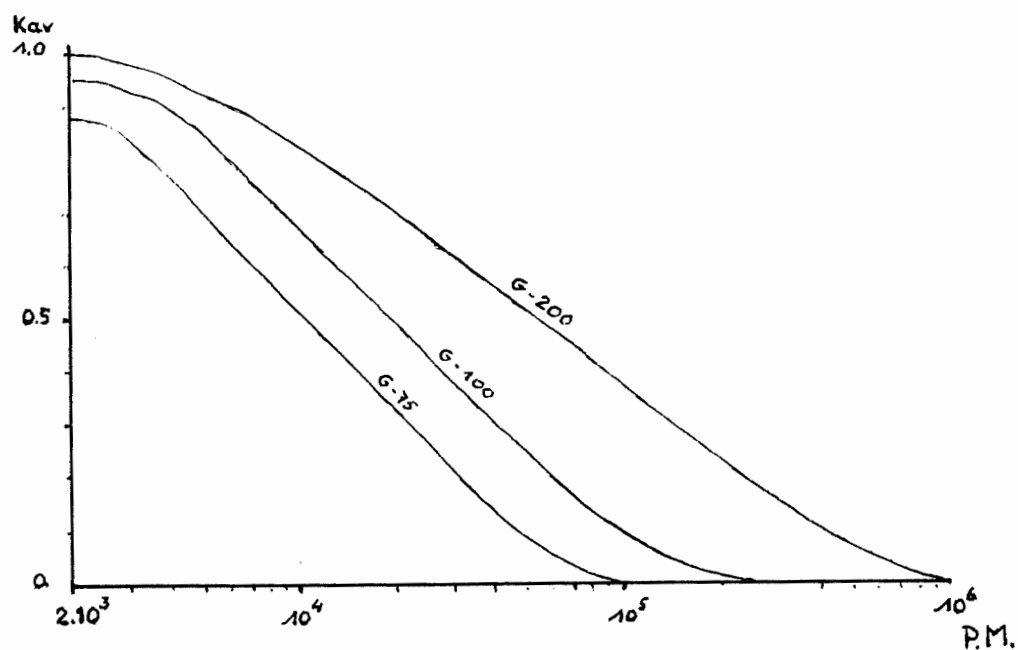
Ce gel est en fait un gel de dextran. Ce dernier est synthétisé à partir du glucose par une bactérie présente dans les dents. Le produit est très hétérogène et est donc purifié, hydrolisé et fractionné. Ce gel est livré sous forme de poudre.

Le tableau 1 ci-dessous donne les propriétés de quelques Sephadex :

type de Sephadex	diamètre des gr.	domaine de fract.	eau abs.	volume du gel	gonflement du gel	
					20°C	100°C
S. G75	40-120	6000- 65000	7.5	12-15	24	3
S. G100	40-120	1000- 100000	10	15-20	72	5
S. G200	40-120	1000- 200000	20	30-40	72	5

où le diamètre des grains est exprimé en microns,
le domaine de fractionnement en dextrans,
la quantité d'eau absorbée en ml/g de gel,
le volume du gel gonflé en ml/g,
le gonflement du gel en heures.

Le choix du type de Sephadex approprié dépend de la dimension des molécules et des propriétés chimiques des substances à séparer. Chaque type de Sephadex sépare dans un domaine de poids moléculaire particulier, déterminé par le taux de gonflement du gel. Les molécules dont le PM est plus élevé que la limite supérieure de ce domaine - limite d'exclusion - sont totalement exclues du gel. Les molécules dont le PM est plus petit que la limite inférieure seront généralement éluées par un volume sensiblement égal au volume du gel.



Figures 4 et 5 :

Variation de l'élution en fonction des propriétés moléculaires: les courbes montrent l'évolution du K_{av} en fonction du PM (en échelle logarithmique) pour des protéines globulaires. [Pharmacia]

Les courbes de la figure 4 donnent la corrélation entre K_{av} et le PM (en échelle logarithmique) pour les trois types de Sephadex.

2°) le gel de polyacrylamide

Ce gel est formé à partir d'acrylamide et de méthylène-bis-acrylamide. Comme le Sephadex, ce gel est inerte et très stable.

3°) le gel d'agarose

L'agar, extrait d'algues marines, est un mélange de polysaccharides linéaires. Il se dissout en chauffant et se redurcit en gel dont le degré de réticulation dépend de la concentration en agarose. Le gel, plus fragile que ses précédents, est livré sous forme hydratée.

Le tableau 2 ci-dessous donne quelques caractéristiques de ces gels.

type de Sepharose	limites de fractionnement (dextrans)	concentration en agarose (%)
Sepharose 2B	100000-50000000	2
Sepharose 4B	40000-15000000	4
Sepharose 6B	10000-3000000	6

La figure 5 présente la sélectivité des courbes du Sepharose 6B, Sepharose 4B, Sephadex G200.

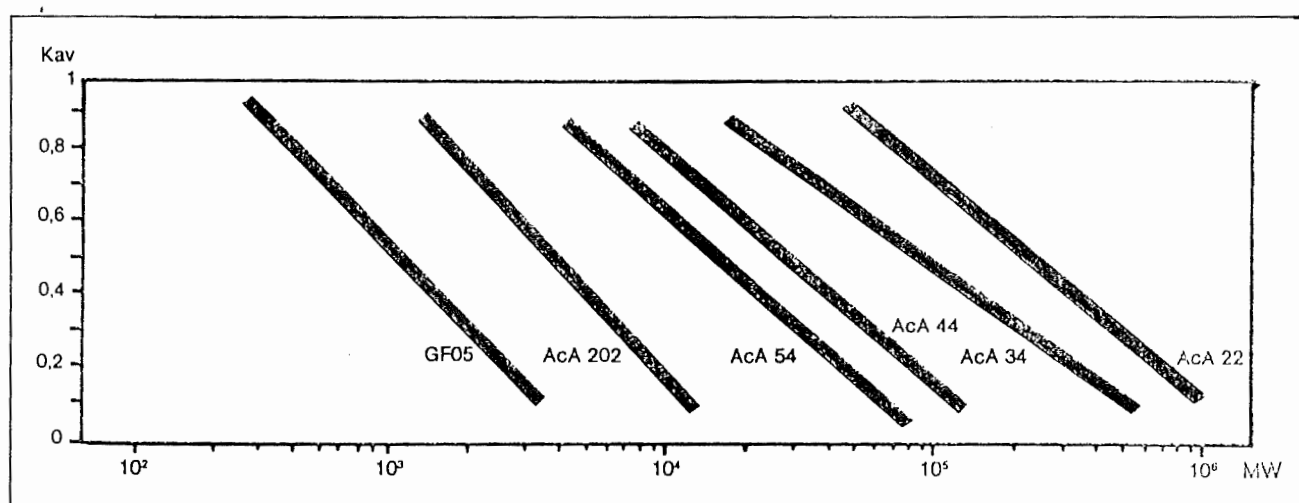


Figure 6 :
 Les courbes montrent l'évolution du Kav en fonction du PM (en
 échelle logarithmique) pour des protéines globulaires.
 [IBF]

4°) les billes de verre poreux

Le verre poreux ayant des pores de taille contrôlée peut servir de support à la chromatographie. Le grand avantage de ce verre est l'inertie chimique et physique ; par exemple, il permet des chromatographies à pH extrêmes, à haute température et sous forte pression. Le seul inconvénient est son prix de revient élevé.

Il existe actuellement toute une série de gels divers vendus sous des appellations variables dépendant des firmes qui les commercialisent. Nous utiliserons par la suite, outre les gels Sephadex et Sepharose, le Sephacryl S200 et les Ultrogel AcA 44 et AcA 22. Les courbes d'étalonnage de ces deux derniers gels sont représentées à la figure 6.

Le Sephacryl est un gel récent provenant de chez Pharmacia-LKB ; il a été optimisé pour atteindre une résolution élevée et des flux rapides. Son domaine de fractionnement pratique se situe entre 5000 et 150000 daltons.

Les Ultrogel AcA sont constitués d'une matrice mixte de polyacrylamide et d'agarose, ce qui permet de combiner la haute résolution du premier et la rigidité du second.

Le tableau 3 ci-dessous donne quelques caractéristiques de ces gels :

type de Sephacryl	limites de fractionnement (dextrans)	diamètre des grains (μ)
Sephacryl AcA 44	10000- 150000	60-140
Sephacryl AcA 22	100000-1200000	60-140

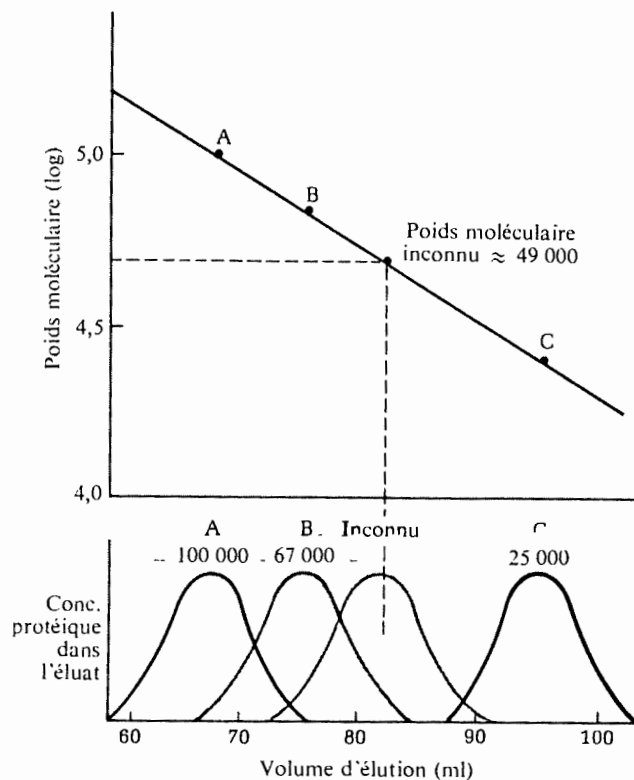


Figure 7 :

Elution des protéines à partir d'une colonne de tamis moléculaire et détermination du poids moléculaire.

Pour étalonner la colonne, deux protéines ou plus (a, b et c) de PM connu sont placées sur la colonne et les volumes d'élution de leurs pics portés sur un graphique en fonction du logarithme de leurs PM. Grâce à son volume d'élution, le PM d'une protéine inconnue peut être calculé à partir du graphique d'étalonnage. [Lehninger A.L., 1977]

a.3. Applications

Les applications de la filtration sur gel sont nombreuses et particulièrement intéressantes.

1°) détermination des masses moléculaires

Nous avons vu que les gels réalisent une séparation en fonction de la masse moléculaire des solutés traités ; les volumes d'élution dépendent de la masse moléculaire. Il devient ainsi possible de tracer la courbe représentant la masse en fonction du volume d'élution, à partir de quelques corps convenables de masse connue. En reportant le volume d'élution du corps étudié, on obtient une valeur approchée souvent largement satisfaisante (voir figure 7).

Cette méthode permet d'étudier en particulier les PM des enzymes contenus dans des milieux complexes.

2°) dessalage

Lorsqu'une solution contient, d'une part, des composés de faible PM et, d'autre part, des composés de PM élevé, le passage sur gel permet de dessaler la solution. Un exemple classique est celui de la préparation de lait sans lactose.

3°) analyse et fractionnement

La méthode de filtration sur gel, très douce, permet la séparation de produits très fragiles, en particulier les protéines et les enzymes. Comme il ne se produit pas de fixation irréversible, l'élution est quantitative.

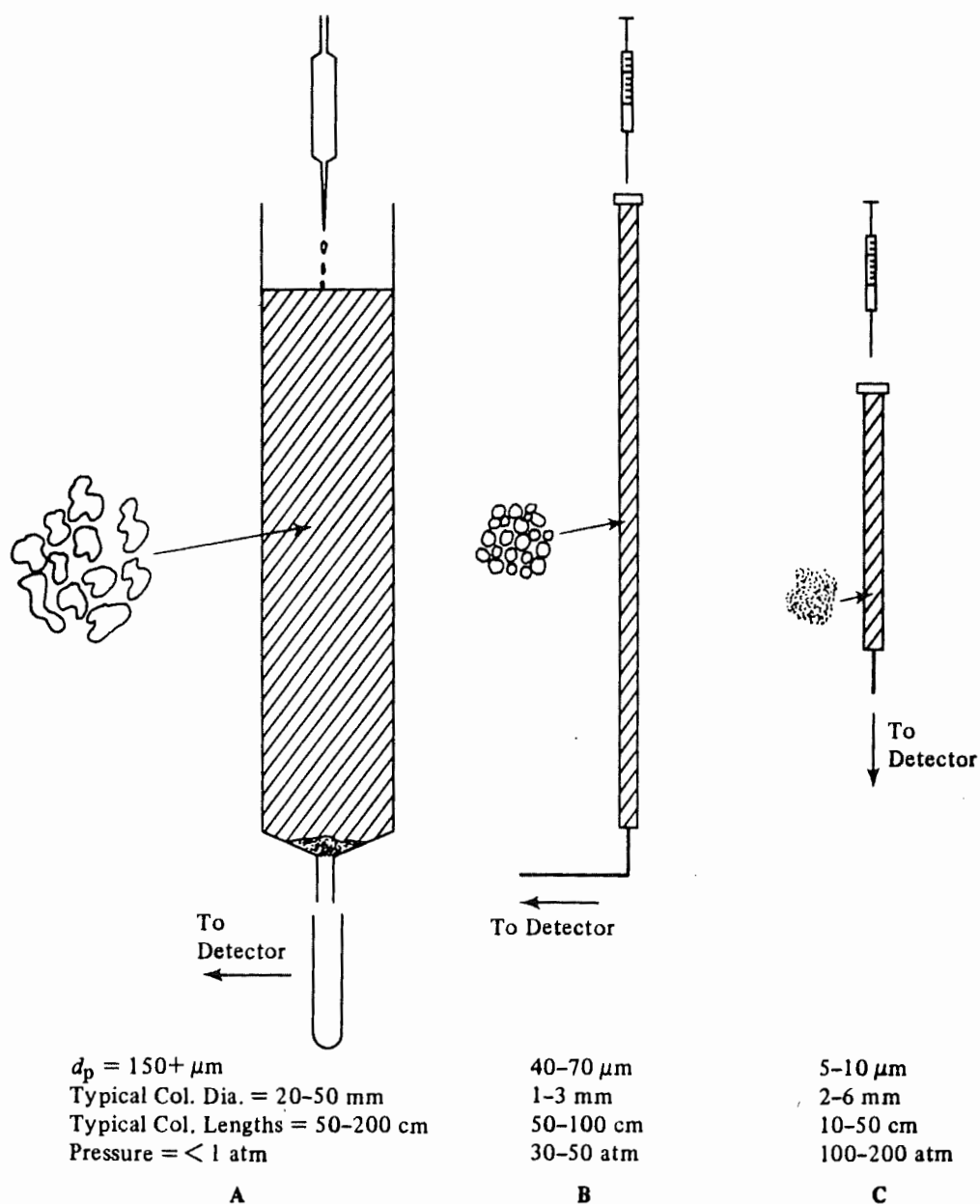


Figure 8 :
 Comparaison de colonnes utilisées en chromatographie liquide :
 A. colonne classique
 B,C. colonnes pouvant être utilisée en HPLC.

b. HPLC ("High Performance Liquid Chromatography")

Afin de mieux comprendre cette technique, mettons en parallèle différentes caractéristiques de la chromatographie liquide et de l'HPLC.

Dans la chromatographie liquide :

- les particules ont de larges pores (100-250 μm)
- le diamètre de la colonne est important (1-5 cm)
- la pression exercée au sommet de la colonne est faible (0.1-1 atm)
- la vitesse d'écoulement est faible (0.1 ml/min)
- le temps de séparation est long.

Dans l'HPLC :

- les particules ont des pores plus petits (5-10 μm)
- le diamètre de la colonne est également réduit (2-6 mm)
- les pressions exercées sont considérables (200 atm)
- la vitesse d'écoulement est plus élevée (1-2 ml/min)
- le temps de séparation est réduit.

La figure 8 résume ces différences.

Cette technique chromatographique est fréquemment utilisée par de nombreux laboratoires pour la purification d'infimes quantités de protéines (de l'ordre de quelques nanomoles). L'hplc permet de réaliser une analyse structurale ou fonctionnelle.

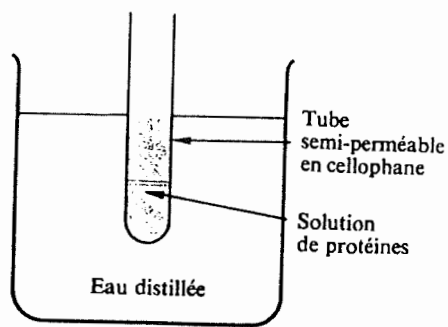


Figure 9 :

Dialyse. La membrane du tube renfermant la solution protéique étant semi-perméable, l'eau et les solutés de faible taille (glucose, NaCl) passent à travers elle, alors que les protéines ne peuvent le faire.
[Lehninger A.L., 1977]

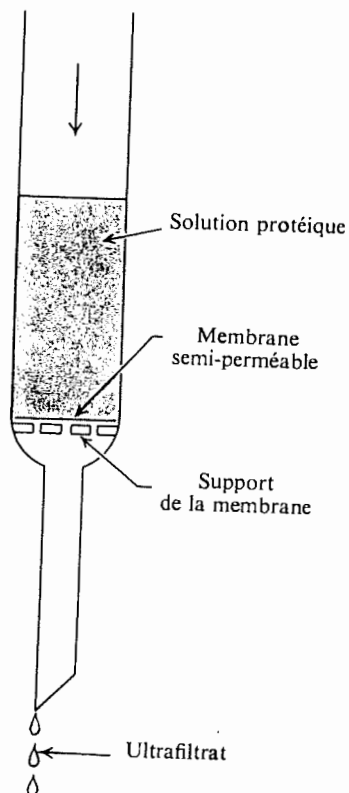


Figure 10 :

Ultrafiltration d'une solution protéique. En appliquant une pression positive au-dessus, la protéine peut être concentrée par filtration de l'eau et des sels dissous.
[Lehninger A.L., 1977]

Les groupements fonctionnels sont distribués finement de telle manière que l'élution peut être accomplie par des tampons qui ne dénaturent que peu les protéines. Les premières colonnes utilisaient de la silice comme support, mais la dissolution de celle-ci sous des conditions alcalines a fortement limité l'utilisation de telles colonnes. Cela a permis le développement de Bio-gels beaucoup plus adaptés.

c. Dialyse et ultrafiltration

Les protéines globulaires en solution peuvent être facilement séparées des solutés de petit poids moléculaire par dialyse (voir figure 9). Cette technique utilise une membrane semi-perméable qui retient les molécules protéiques et permet aux molécules de solutés de petite taille et à l'eau de passer à travers la membrane.

Une autre méthode de séparation des protéines des petites molécules est l'ultrafiltration dans laquelle une pression ou une force centrifuge est utilisée pour filtrer les petites molécules de solutés à travers une membrane semi-perméable qui retient les molécules protéiques (voir figure 10). Dans ces procédés, le cellophane et d'autres matériaux synthétiques sont habituellement utilisés comme membrane.

d. Centrifugation en gradient de densité

En solution, les protéines ont tendance à sédimenter lorsqu'on les soumet à des forces centrifuges élevées. Il est donc possible de séparer un mélange de protéines par des méthodes de centrifugation.

Un gradient de densité de saccharose est d'abord préparé dans un tube à centrifuger par un système qui mélange une solution concentrée de saccharose à de l'eau dans un rapport décroissant au fur et à mesure que le tube est rempli, de telle sorte que la densité du milieu soit la plus grande à l'extrémité inférieure du tube. Le mélange de macromolécules est déposé à la partie supérieure du gradient. La centrifugation à grande vitesse amène chaque type de macromolécules à sédimenter à une position qui dépend essentiellement du poids de la particule, mais aussi de sa densité et de sa forme, ce qui se traduit par des bandes séparées. Les positions de celles-ci peuvent être localisées et leur contenu recueilli et analysé.

3.2.2. Procédés de séparation basés sur la solubilité

La solubilité des protéines en solution est fonction du pH, de la force ionique, des propriétés électriques du solvant et de la température. Ces variables peuvent être utilisées pour séparer des mélanges de protéines.

a. La précipitation isoélectrique

a.1. Principe et théorie

La solubilité de la plupart des protéines globulaires est profondément influencée par le pH du milieu. Presque toutes les protéines ont une solubilité minimale pour un pH qui diffère d'une protéine à l'autre ; de part et d'autre de ce pH critique, la solubilité augmente considérablement. Le pH de moindre solubilité d'une protéine est son pH isoélectrique, défini comme le pH pour lequel la molécule n'a pas de charge électrique nette et est ainsi incapable de migrer dans un champ électrique. Dans

ces conditions, il n'existe pas de répulsion électrostatique entre les molécules protéiques voisines et elles ont tendance à précipiter. Pour des pH supérieurs ou inférieurs au pH isoélectrique, toutes les molécules protéiques ont une charge nette de même signe. Elles vont ainsi se repousser les unes les autres, empêchant de ce fait la coalescence de molécules isolées en agrégats insolubles.

En augmentant la concentration en sels dans un milieu, la force ionique μ augmente:

$$\mu = 1/2 \sum C_i Z_i^2$$

où C_i = concentration d'un ion i
 Z_i = charge de l'ion i .

L'addition de faibles quantités de sels a comme effet de neutraliser partiellement les charges des protéines et de diminuer le potentiel chimique des protéines en solution ; ce qui amène une augmentation de la solubilité. A force ionique plus forte, les ions vont s'hydrater et entrer en compétition avec les protéines pour fixer l'eau d'hydratation. Dans ces conditions, le logarithme de la solubilité diminue avec la force ionique :

$$\log S = B - (K * \mu).$$

Ceci représente l'équation d'une droite, où K est la pente de la droite (très semblable pour un même agent de précipitation) et B la constante de solubilité (variant en fonction de la nature de la substance).

Si l'opérateur connaît les valeurs de K et de B , il pourra estimer la force ionique nécessaire pour que la protéine qu'il souhaite purifier précipite. Remarquons cependant que de telles valeurs sont rarement disponibles dans la littérature.

a.2. Application

Les protéines ont des pH isoélectriques qui diffèrent selon leur contenu en acides aminés possédant des chaînes latérales ionisables ; elles peuvent donc être séparées les unes des autres par une précipitation isoélectrique. Quand le pH d'un mélange protéique est ajusté au pH isoélectrique de l'un de ses composants, la totalité de ce composant précipite, laissant en solution les protéines dont le pH isoélectrique est supérieur ou inférieur au pH choisi.

Le principe de la technique est donc simple : le mélange de protéines à purifier est dilué et versé dans un récipient.

L'opérateur ajoute progressivement un agent de précipitation, souvent une solution de sulfate d'ammonium saturée à 4°C et de pH 7.4. La force ionique du milieu augmente donc progressivement. Sachant que les protéines du mélange précipitent à différentes valeurs de force ionique, il est donc possible de les séparer sur cette base.

a.3. Critiques

Un avantage notable de la méthode est que les protéines précipitées conservent leur conformation native et peuvent être redissoutes dans un milieu qui possède un pH et une concentration saline appropriés.

De plus, cette technique est simple à utiliser. Elle ne permet cependant pas de purifier parfaitement une protéine ; elle la concentre. Elle constitue donc une méthode complémentaire aux chromatographies.

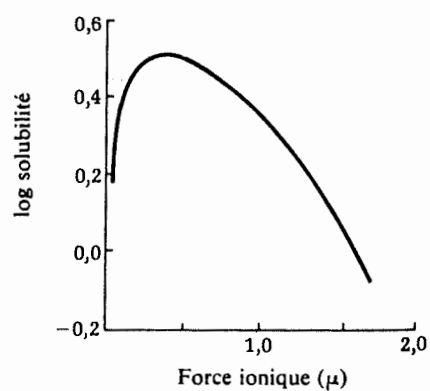


Figure 11 :

A basse force ionique, la solubilité de la protéine augmente (salting-in). A concentration en sels élevée, elle diminue (salting-out ou relargage).

[Lehninger A.L., 1977]

La précipitation est particulièrement intéressante lorsque la quantité de matériel à purifier est importante (du gramme au kilo). Une première purification peut dans ce cas être réalisée rapidement en utilisant un minimum de moyens.

b. Dissolution par les sels et relargage des protéines

Les sels neutres agissent sur la solubilité des protéines globulaires de la façon suivante : à faible concentration, les sels augmentent la solubilité de nombreuses protéines ; ce phénomène est appelé "salting-in" ou dissolution par les sels. L'action des sels neutres sur la solubilité des protéines est fonction de leur force ionique. L'effet de "salting-in" est dû à des modifications de la tendance spontanée d'ionisation des chaînes latérales dissociables sur la protéine (voir figure 11).

Par ailleurs, lorsque la force ionique augmente, la solubilité d'une protéine commence à diminuer ; à une force ionique suffisamment élevée, une protéine peut être (presque) complètement précipitée : cet effet est appelé "salting-out" ou relargage. La base physico-chimique du relargage est complexe.

L'intérêt de la méthode provient du fait qu'elle permet de séparer un mélange protéique, puisque les protéines ne répondent pas toutes de la même façon à des concentrations de sels neutres. De plus, les protéines précipitées conservent leur conformation native.

c. Fractionnement par des solvants

L'addition de solvants organiques neutres, miscibles à l'eau, diminue la solubilité dans l'eau de la plupart des protéines globulaires, de telle sorte qu'elles peuvent précipiter dans la solution.

Cette méthode est basée sur le fait que la solubilité protéique à un pH et une force ionique donnés est fonction de la constante diélectrique du milieu. L'addition d'un solvant de constante diélectrique plus faible que celle d'une solution aqueuse contenant des protéines augmente les forces attractives entre les charges opposées et les protéines tendent alors à s'agréger et à précipiter.

d. Effet de la température sur la solubilité des protéines

Dans un écart limité de température (0°-40°C), la plupart des protéines globulaires ont une stabilité qui diminue avec la température. Les procédés de fractionnement sont effectués à 0°C, puisque la majorité des protéines sont stables à basse température.

3.2.3. Procédés de séparation basés sur la charge électrique

La séparation des protéines selon leur charge électrique dépend en fin de compte de leurs propriétés acido-basiques qui sont largement déterminées par le nombre et le type des chaînes latérales ionisables de leurs chaînes polypeptidiques. Les protéines diffèrent par leur composition et leur séquence en aminoacides et chacune d'elles possède ainsi des propriétés acidobasiques distinctes.

a. La chromatographie sur échangeur d'ions

a.1. Principe et théorie

Les échangeurs d'ions sont des composés ioniques solides qui peuvent, lorsqu'ils sont en présence d'une solution, échanger une partie des ions qu'ils portent contre ceux de la solution. Dans des conditions bien déterminées de pH ou de force ionique, ils pourront fixer des ions de la solution qui les baigne et les relâcher quand ces conditions seront différentes (c'est-à-dire lors de la phase d'élution). Il est donc possible, avec de tels échangeurs d'ions, de pratiquer des analyses de type chromatographique.

On distingue couramment deux types d'échangeurs : les échangeurs d'anions et les échangeurs de cations. Les premiers ont des groupements fonctionnels positifs, il s'agit d'ions ammonium -NH_3^+ , >NH_2^+ , >NH^+ , $\text{N}^+ \llcorner$.

Les seconds contiennent des groupements négatifs de type -COO^- , -SO_3^- , $\text{-CH}_2\text{-SO}_3^-$, ...

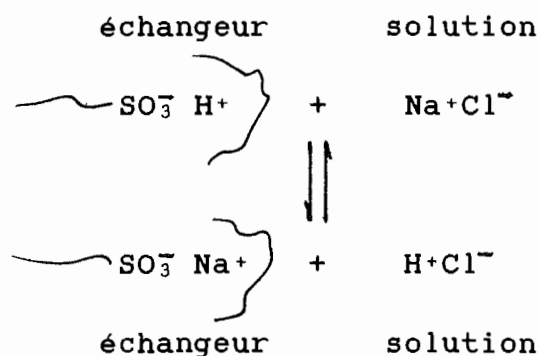
(Les '-' correspondent à des liaisons covalentes.)

Exemple : soit une résine échangeuse de cations se trouvant sous forme H^+ , et l'on ajoute une solution de NaCl contenant également des ions H^+ . Au début, les ions Na^+ vont remplacer certains ions H^+ de la colonne et ceux-ci vont diffuser dans la phase liquide. Au fur et à mesure que le solvant est ajouté à la colonne, un équilibre va s'établir entre les ions Na^+ et H^+ pour se fixer sur les radicaux négatifs et l'on aura

$$K = \frac{\frac{[H^+]_l}{[Na^+]_l}}{\frac{[H^+]_c}{[Na^+]_c}}$$

avec l = dans le liquide
c = obtenu sur la colonne

K est le coefficient d'équilibre ou de partage entre ces deux ions.



a.2. Différents types d'échangeurs

Les échangeurs minéraux d'origine naturelle, tels que les argiles, certains micas, furent les premiers échangeurs utilisés.

Actuellement, on utilise les gels échangeurs d'ions. Ces derniers sont utilisés préférentiellement pour séparer des macromolécules qui se comportent comme des poly-électrolytes (les protéines, les acides nucléiques,...). Les deux groupes fonctionnels les plus utilisés sont le DiEthylAminoEthyle (DEAE), le CarboxyMéthyle (CM) et la SulfoPropyle (SP).

Ces échangeurs ont une capacité élevée et une adsorption non spécifique faible, ce qui les rend très utiles, en particulier en biochimie.

a.3. Elution

L'élution d'un composé fixé sur un échangeur s'obtiendra par tous les procédés qui feront varier l'ionisation de l'échangeur.

Un premier procédé consiste à modifier le pH : en effet, en amenant le pH vers le pI de la protéine, on diminue sa charge et donc sa fixation.

Un second procédé consiste à augmenter la force ionique (μ) de la solution à l'aide d'un ion assez "fort" pour déplacer les composés fixés sur l'échangeur et sous une concentration suffisante pour modifier l'équilibre de la relation d'échange. Ces modifications peuvent se réaliser par étapes successives ou par gradient.

a.4. Applications

1°) transformation d'un constituant ionique d'une solution

C'est la technique employée dans l'adoucissement de l'eau : l'eau "dure" chargée en sels de calcium, passe sur un échangeur qui fixe l'ion calcium, en libérant en échange des ions sodium. Le sel de calcium a été transformé en sel de sodium, qui ne présente plus les propriétés entartrantes du sel de calcium. La régénération s'effectue par traitement de l'échangeur par une solution concentrée de chlorure de sodium.

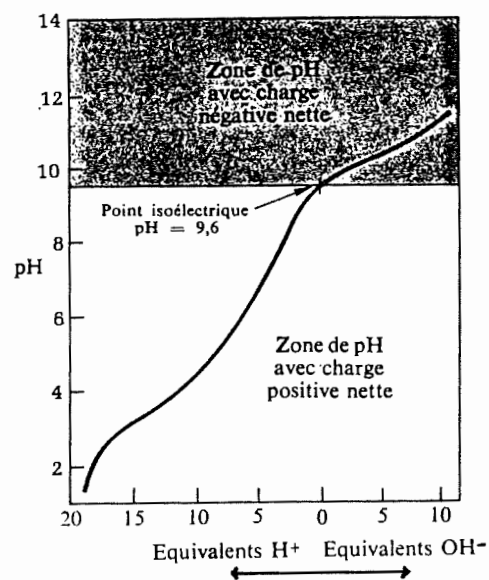


Figure 12 :
 Courbe de titrage de la ribonucléase à partir du point
 isoélectrique de celle-ci qui vaut 9.6 pH.
 [Lehninger A.L., 1977]

2°) élimination d'impuretés ioniques

C'est le cas de la déminéralisation totale de l'eau, dans laquelle on fait passer l'eau brute sur deux types de résines anioniques et cationiques.

b. Les méthodes électrophorétiques

Comme on l'a défini antérieurement, l'électrophorèse est basée sur la migration de particules chargées en solution dans un champ électrique.

Nous allons maintenant décrire cette technique de manière plus détaillée.

b.1. Principe de base

Les ions, particules provenant de la dissociation d'un électrolyte dissout dans l'eau, vont permettre la conduction. Dès lors, les cations (ions chargés positivement) migreront vers le pôle négatif et les anions (chargés négativement) vers le pôle positif.

La charge d'un composant est fonction du pH de la solution. En effet, tout ampholyte (molécule comprenant plusieurs charges) est chargé positivement à des valeurs de pH inférieures à son point isoélectrique et négativement si le pH est supérieur à son point isoélectrique ; dans le cas où le pH est égal au pH isoélectrique, l'ampholyte est neutre et donc ne migre pas dans un champ électrique (voir figure 12).

Après avoir précisé ces concepts théoriques, il est plus aisé de comprendre la technique d'électrophorèse proprement dite.

Cette technique consiste à injecter un échantillon d'un mélange dont on veut séparer les constituants dans une région du support (gel en tube ou plaque, notions que nous détaillerons plus loin) ; ensuite, le gel est immergé dans un tampon adéquat et un champ électrique est appliqué grâce à l'introduction de deux électrodes placées de part et d'autre du gel. Les constituants migrent alors en fonction de leur charge nette, c'est-à-dire le défaut ou l'excès d'électrons, de la même façon que le font les ions d'un électrolyte pendant l'électrolyse.

b.2. Types d'électrophorèse

Les techniques électrophorétiques essentielles peuvent se diviser en deux grands groupes : (1) électrophorèse en phase libre ou par gradient mobile dans laquelle la migration des particules se fait librement dans un liquide donné ; (2) électrophorèse dans des milieux qui s'opposent à la convection ou électrophorèse de zone dans laquelle la migration des particules se fait dans un liquide soutenu par un support approprié.

La première méthode, bien qu'elle soit d'une grande utilité dans la détermination des valeurs absolues de mobilité des différents composants, ne se prête pas à des applications sur une grande échelle, en ce sens qu'elle nécessite une quantité notable de matériel, un temps d'exécution assez long et qu'elle ne permet pas des déterminations de propriétés spécifiques pour les différents constituants ; enfin, on ne peut l'utiliser dans un but préparatif.

Ces principaux désavantages ne se retrouvent heureusement pas dans la deuxième méthode qui, pour cette raison, est la plus souvent utilisée. Les différents milieux employés pour l'électrophorèse de zone seront décrits dans la section suivante.

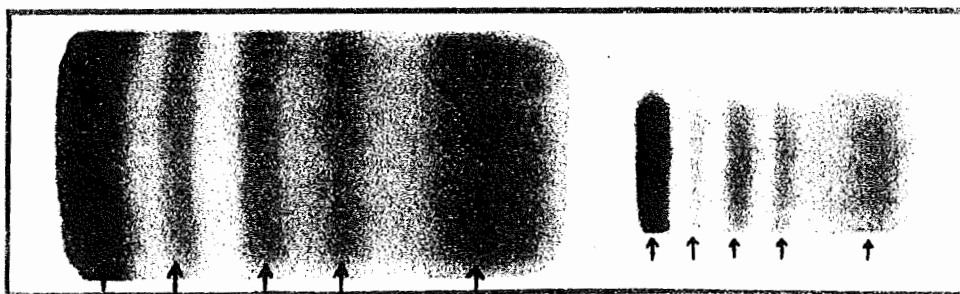


Figure 13 :
 Traces électrophorétiques (à gauche sur papier, à droite sur
 acétate de cellulose) des protéines du sérum humain normal.
 [Encyclopédie int., 1971]

Cette méthode possède une résolution élevée ; c'est pourquoi elle est souvent utilisée dans les laboratoires cliniques pour doser les protéines sanguines.

b.3. Supports utilisés en électrophorèse

Comme nous l'avons souligné auparavant, dans l'électrophorèse de zone, la solution aqueuse est immobilisée dans une matrice solide, matériel poreux hydraté qui possède une rigidité mécanique éliminant les anomalies dues à la convection ou à la vibration et ne retarde pas les mouvements des particules.

Les supports les plus largement utilisés sont le papier filtre et l'acétate de cellulose, matériaux qui n'interagissent pas avec les protéines migrantes (voir figure 13). L'électrophorèse se poursuit jusqu'à ce que les principaux composés protéiques se séparent en zones. La quantification de protéines est possible grâce à une coloration de celles-ci : leur densité est proportionnelle à l'intensité de la coloration. Ce support est intéressant pour de nombreuses applications, mais en raison du fait que les particules migrantes sont adsorbées à la surface des fibres de cellulose, il se produit des anomalies au niveau de la conduction des ions. L'augmentation de la température dans les zones de résistance électrique assèche le papier et perturbe la migration.

Pour ces raisons, des gels d'amidon ou de polyacrylamide sont préférés. Ces gels sont stabilisés par de faibles liens intermoléculaires ; ils doivent être électriquement neutres et résistants afin de pouvoir être manipulés aisément. Ce type de support permet de séparer un mélange selon à la fois sa charge électrique et son poids moléculaire.

Les gels formés uniquement d'acrylamide présentent des propriétés non désirées ; de plus la taille des pores, facteur déterminant pour une résolution optimale, est mal définie (la taille moyenne des pores dépend du rapport entre le poids d'acrylamide et le volume total du gel). Dès lors, il est plus intéressant d'y ajouter une faible proportion de bisacrylamide.

Ces types de support présentent de nombreux avantages dont celui de pouvoir être utilisés pour effectuer l'isolement de grandes quantités de protéines purifiées.

b.4. Electrophorèse sur gel avec SDS

De nombreuses protéines contiennent plus d'une chaîne polypeptidique. Grâce à une variante de l'électrophorèse de zone appelée électrophorèse sur gel-SDS, une protéine oligomérique peut être dissociée en sous-unités, leur nombre et leur poids moléculaire pouvant ensuite être déterminés. La méthode peut être utilisée également pour découvrir le poids moléculaire des protéines monomériques.

Le principe est relativement simple : la protéine purifiée est traitée avec un détergent, le sodium dodécylsulfate (ou SDS), qui la dissocie en sous-unités et déroule complètement chaque chaîne polypeptidique pour former un complexe allongé entre le polypeptide et le SDS.

Dans ce complexe, la chaîne polypeptidique est recouverte d'une couche de molécules de SDS dont les chaînes hydrocarbonées hydrophobes sont solidement liées à la chaîne polypeptidique ; ainsi, les groupements chargés du SDS sont exposés au milieu aqueux.

Dans ces complexes, il existe normalement un rapport constant entre SDS et protéines ; ils ne diffèrent donc que par leur masse.

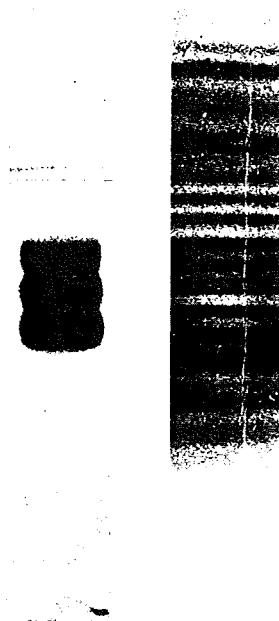


Figure 14 :

L'électrophorèse discontinue sur gel (à gauche) sépare les enzymes en seulement trois bandes, tandis que l'électrofocalisation (à droite) sépare 18 formes moléculaires différents de l'enzyme.

[Lehninger A.L., 1977]

Afin de pouvoir déterminer le poids moléculaire de protéines inconnues, il est nécessaire de calibrer le gel au moyen de protéines marqueuses connues soumises au même processus.

Malheureusement, la méthode ne permet pas de chiffrer le poids moléculaire de toutes les protéines. En effet, certaines glycoprotéines migrent plus lentement que ce à quoi on pourrait s'attendre à la vue de leur PM.

Il est utile de préciser que l'utilisation de plusieurs techniques en combinaison permet d'obtenir des moyens d'analyse de protéines plus complexes et plus efficaces. Il est dès lors conseillé de ne pas se braquer sur une seule technique.

b.5. Electrofocalisation

Le processus le plus ingénieux et le plus efficace pour séparer des protéines est la focalisation isoélectrique ou électrofocalisation, où un mélange de protéines est soumis à un champ électrique dans un gel support dans lequel un gradient de pH a été préalablement formé.

Chaque protéine va migrer et se localiser dans la portion du gradient où le pH est identique à son pH isoélectrique, et former ainsi une bande étroite stationnaire.

Le pouvoir de résolution de l'électrofocalisation est extraordinaire (voir figure 14). Cette technique, normalement utilisée comme méthode analytique, peut également être employée à une plus grande échelle pour préparer des protéines purifiées.

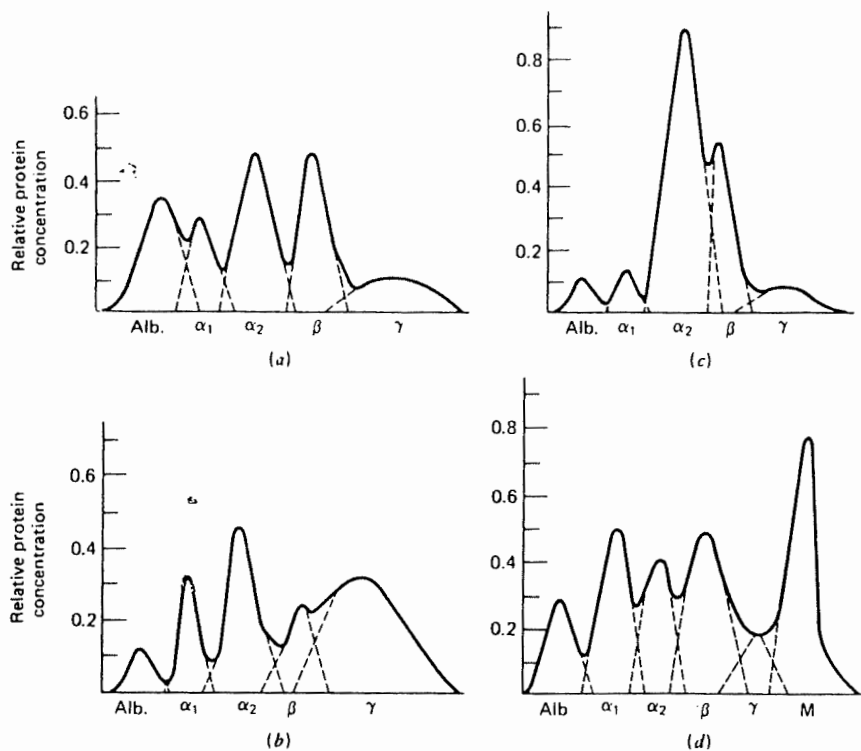


Figure 15 :

Profils électrophorétiques sur acétate de cellulose de

(a) sérum normal

(b) sérum d'individu souffrant d'une hépatite infectieuse

(c) sérum d'individu souffrant d'une 'néphrose lipidique'

(d) sérum d'individu souffrant d'un myélome gamma.

b.6. Applications

Bien que la biochimie soit un domaine qui utilise énormément les techniques d'électrophorèse, d'autres disciplines en font également usage. En effet, de nombreux diagnostics médicaux reposent sur la séparation de constituants de fluides biologiques qui peuvent ainsi être analysés individuellement.

L'élément biologique qui est le plus étudié est le sang et particulièrement les protéines plasmatiques. En comparant les résultats obtenus après une électrophorèse de protéines sanguines d'un individu sain avec ceux d'un individu malade, il est possible de déceler des anomalies, et donc d'établir un diagnostic (voir figure 15).

Mais il n'y a pas que le sang qui présente un intérêt en médecine ; le liquide cébrospinal et l'urine sont également étudiés. Cependant, leur analyse est rendue difficile par le fait que les protéines constitutives de ces "mélanges" sont en concentration très faible.

Des variations de quantités de protéines du liquide cébrospinal peuvent être la conséquence de multiples maladies telles que la sclérose, la méningite, etc.

Quant à des modifications de concentration de protéines urinaires, elles peuvent indiquer un mauvais fonctionnement rénal.

Il est également intéressant de noter que les composants de l'hémoglobine peuvent être séparés au moyen de la technique d'électrofocalisation.

Une variante de l'électrophorèse, l'immuno-électrophorèse, permet de déterminer la concentration inconnue jusqu'alors d'un antigène mis en présence de son anticorps.

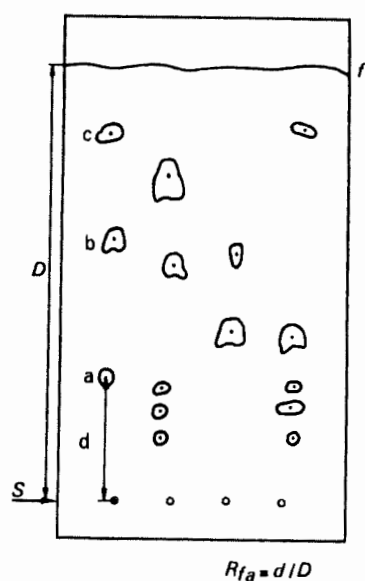


Figure 16 :

Résultat de la chromatographie sur papier :

- S = départ
- D = distance du départ au front
- d = distance du départ au centre de la tache a.

3.2.4. Autres types de chromatographie

a. Chromatographie sur papier

La chromatographie sur papier est la technique chromatographique la plus simple et était la plus utilisée avant l'apparition de la chromatographie sur couche mince.

Cette technique peut être considérée comme une chromatographie d'adsorption entre un solvant liquide mobile et une phase solide immobile. On peut également imprégner le papier d'un solvant (habituellement l'eau) et faire migrer un autre solvant, nous aurons alors une chromatographie de partage. Suivant l'affinité d'une substance pour le support solide et pour la phase mobile, on observera une migration plus ou moins rapide de cette substance avec le solvant.

Il existe deux techniques principales de chromatographie sur papier : l'ascendante et la descendante. Pour chaque substance chromatographiée sur un support donné et dans un solvant (ou mélange) déterminé, on définit une valeur :

$$r_f = \frac{\text{distance parcourue par la substance (= d)}}{\text{distance parcourue par le solvant (= D)}}$$

(voir figure 16)

Le r_f est une valeur caractéristique d'un composé pour des conditions de chromatographie données (papier, solvant, température, ...).

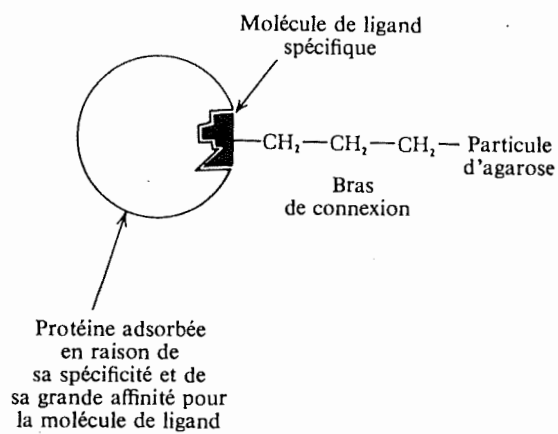


Figure 17 :
Principe de la chromatographie d'affinité.
 [Lehninger A.L., 1977]

b. Chromatographie d'affinité

Il s'agit d'une méthode récente de séparation basée non plus sur des caractères physico-chimiques (solubilité, pH, charge électrique) mais utilisant au contraire des interactions spécifiques ; par exemple, liaison enzyme-substrat ou antigène-anticorps (voir figure 17).

La molécule (ou ligand) possédant une affinité particulière pour la protéine à purifier est fixée de manière covalente sur un support solide (gel, résine). Le mélange de protéines est mis en contact avec ce gel, soit sur une colonne, soit en tube. Le gel est lavé et seule la protéine ayant une affinité suffisante pour le ligand va rester adsorbée. Après ce lavage, celle-ci est éluée en modifiant le milieu d'élution par la présence du substrat, le pH ou la force ionique. Malheureusement, les protéines n'ont pas toujours un ligand spécifique disponible ou bien connu et ceci explique les raisons pour lesquelles cette technique n'est pas systématiquement abordable.

La chromatographie d'affinité est utilisée pour isoler non seulement les enzymes, mais aussi des récepteurs dans les membranes cellulaires qui fixent spécifiquement les hormones.

c. Chromatographie sur couche mince

Il s'agit d'une application particulière de la chromatographie d'adsorption.

Un adsorbant, différent selon les besoins, est étalé sur une plaque de verre. Après une éventuelle activation par la chaleur suivie du dépôt des substances à analyser, la plaque est placée dans une cuve contenant le solvant. La migration terminée, les résultats seront observés en utilisant différentes méthodes de révélation.

Elle est utilisée principalement comme outil analytique pour séparer et doser de petites substances lipophiles (stéroïdes, lipides,...). Elle est de plus en plus remplacée par la chromatographie en phase gazeuse.

d. Chromatographie en phase gazeuse

Cette technique a été tout d'abord un type particulier de chromatographie d'adsorption, où la phase mobile est gazeuse. Cette méthode de séparation est donc applicable aux gaz et à tous les corps volatilisables. La phase mobile est ici un gaz inerte, qui entraîne les substances dans un fin tuyau (colonne) où se trouve la phase stationnaire.

Lorsque cette phase immobile est un solide, on parlera de chromatographie gaz/solide (C.G.S.) et la séparation des diverses substances se fera suivant les propriétés d'adsorption des substances.

Lorsque cette phase solide est recouverte d'un liquide, on parlera de chromatographie gaz/liquide (C.G.L.) et le principe de séparation sera celui de la chromatographie de partage. Le développement rapide de la C.P.G. s'explique par la rapidité, la sensibilité et le grand nombre d'applications de cette technique d'analyse et de dosage.

e. Conclusion

De ce coup d'oeil sur la chromatographie, on peut tirer quelques remarques : notons tout d'abord la grande diversité des principes qui entrent en jeu (voir tableau 4). De plus, les performances séparatives de la chromatographie dépendent d'un certain nombre de paramètres.

Le premier à prendre en considération est la nature de la protéine à séparer et ses caractéristiques intrinsèques. En fonction des caractères prédominants, il sera d'abord possible de choisir le type de chromatographie le plus adapté ou les meilleures combinaisons chromatographiques.

Le deuxième paramètre à considérer est le degré de "contamination" de la protéine à séparer (concentration relative par rapport aux autres protéines à éliminer).

Le troisième paramètre important est la concentration de la protéine dans le milieu ainsi que le pH et la force ionique de ce dernier.

PRÉCIPITATION	Insolubilité de certaines protéines sous l'action de produits chimiques déterminés	Concentration Séparation par groupes de protéines
ULTRAFILTRATION	Différence de masse moléculaire de composants d'un mélange	Séparation de macro-molécules & micro-molécule Concentration Dessalage
ÉLECTROPHORÈSE	Différence de mobilité (charge et/ou éventuellement, masse moléculaire) des composants du mélange soumis à un champ électrique	Séparation analytique par groupe de protéines Détermination de masses moléculaires et de pI Concentration dans certains cas
CHROMATOGRAPHIE	Différence de masse moléculaire Différence de charge nette Différence de degré d'hydrophobicité Différence de densité de charge Possibilité de former des complexes spécifiques Activité biologique Point isoélectrique	Séparation par famille de protéines Dessalage Concentration Séparations bio-spécifiques Détermination de masses moléculaires

Tableau 4 :
Les principales techniques séparatives appliquées aux protéines.
[Boschetti E., 1987]

II. CONTEXTE INFORMATIQUE

II. CONTEXTE INFORMATIQUE

Dans le même ordre d'idée que le chapitre premier, nous allons vous donner un aperçu du domaine informatique sur lequel est basé notre travail et ce, dans le but de familiariser le lecteur aux concepts de l'intelligence artificielle et des systèmes experts.

1. Introduction à l'intelligence artificielle

1.1. Définition et domaines de l'intelligence artificielle

L'informatique est entrée depuis longtemps dans nos mœurs et est désormais bien intégrée dans notre vie quotidienne. Au sein des entreprises, une bonne informatique est devenue un élément vital. Dès qu'un travail peut être réduit à un algorithme clairement défini, il est dorénavant confié à un ordinateur. Ce concept d'algorithme qui sous-tend une organisation prédéfinie et immuable dans le temps caractérise tous les programmes de gestion conventionnels. Par contre, ce type de programme est incapable de résoudre des problèmes qui seraient caractérisés par une composante variable ou qui n'aurait pas été prédéfinie. La naissance de l'intelligence artificielle a permis d'envisager la résolution de problèmes de ce genre.

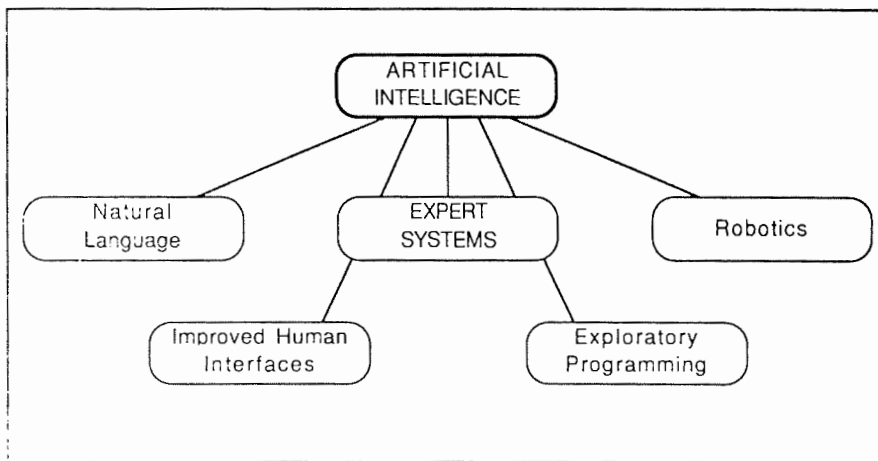


Figure 18 :
L'arbre généalogique de l'intelligence artificielle.
[P.Harmon et al., 1988]

Cette discipline connaît depuis une dizaine d'années des développements importants, tant sur le plan de la recherche que sur celui des applications.

Ainsi, P.Harmon, R.Maus et W.Morrissey (Expert Systems Tools and Applications) admettent que l'intelligence artificielle se divise en cinq champs d'activité : (1) la compréhension du langage naturel, (2) la robotique avancée, (3) l'amélioration des interfaces homme-machine, (4) la programmation d'exploration et (5) les systèmes experts. La figure 18 présente ces différents secteurs d'activité. Nous allons introduire brièvement les quatre premiers domaines, le dernier faisant l'objet d'un développement ultérieur.

1.1.1. Le langage naturel

Un langage naturel est un langage quelconque parlé par l'homme comme, par exemple, l'anglais ou le français. Les recherches développées en intelligence artificielle tendent à mettre au point des technologies hardware et software qui permettront aux ordinateurs de dialoguer avec l'homme dans un langage naturel. Par conséquent, avec ce type d'outil, un utilisateur pourrait rechercher de l'information dans une base de données en formulant une demande écrite ou orale de la même manière qu'il le ferait avec son assistant.

1.1.2. La robotique avancée

Créer des mécanismes robotiques constitue une des préoccupations majeures des chercheurs en intelligence artificielle. Par définition, le robot peut agir "intelligemment" étant donné qu'un modèle du monde réel est stocké dans l'ordinateur. Cela permet à ce dernier d'identifier

des objets et d'analyser comment ces objets vont changer suite aux diverses actions réalisées par le robot.

1.1.3. L'amélioration des interfaces homme-machine

Un troisième domaine dans lequel les concepts et techniques utilisés en intelligence artificielle ont été activement employés consiste en la conception et le développement de meilleures interfaces (exemple : Macintosh). Ces mêmes techniques hardware et software qui ont rendu le Mac plus convivial et populaire sont maintenant présentes sur les nouvelles stations de travail et sur les PC's (exemple : Windows).

1.1.4. La programmation explorative

Ce concept se réfère à l'application des concepts et techniques utilisés en intelligence artificielle au développement de grandes applications. La "programmation automatique" peut également faire partie de ce quatrième domaine d'application. Cette méthode de programmation utilise des techniques d'intelligence artificielle pour permettre aux programmeurs de développer des logiciels en spécifiant les objectifs de ces derniers. Par la suite, un "système de programmation automatique" générera le code spécifique à l'application.

Comme il a été dit précédemment, le dernier domaine d'application de l'intelligence artificielle, à savoir les systèmes experts, est celui qui a retenu le plus l'attention des chercheurs et il sera traité ultérieurement.

1.2. Les idées clés de l'intelligence artificielle

La rédaction de ce paragraphe est basée sur l'ouvrage de P.Harmon, R.Maus et W.Morrissey : Expert Systems Tools and Applications.

Durant ces dernières années, un certain nombre de concepts ont été développés par les laboratoires de recherche en intelligence artificielle. Les trois idées majeures qui émergent de ces études sont :

- de nouvelles manières pour représenter les connaissances,
- la recherche heuristique,
- la séparation entre les connaissances relatives au problème et celles de contrôle.

Nous allons brièvement développer chacune de ces idées.

1.2.1. La représentation des connaissances

Il est admis que la connaissance est plus complexe que l'information et surtout plus précieuse. Le terme connaissance se réfère à un corps d'informations concernant un sujet particulier et qui est organisé de manière à être exploité.

Les chercheurs en intelligence artificielle se sont focalisés sur les aspects verbaux et graphiques de la connaissance plutôt que sur ses aspects mathématiques. Par conséquent, quand un programmeur "conventionnel" cherche à réduire un problème en éléments pouvant être représentés en termes mathématiques et donc manipulés par un algorithme, les programmeurs en intelligence artificielle sont plus intéressés

par la connaissance exprimée en phrases et schémas, et manipulée par des inférences logiques.

Dès lors, si l'usage de l'ordinateur continue à croître et à fournir de l'aide aux utilisateurs qui sont confrontés à des problèmes de plus en plus complexes, les programmes informatiques devront être capables d'aider à l'analyse et à la résolution de problèmes qui ne seront plus exprimés qu'en termes linguistiques.

1.2.2. La recherche heuristique

La programmation "conventionnelle" est basée sur une analyse exhaustive de tous les éléments et de toutes les étapes conduisant à la résolution d'un problème. Par conséquent, ceci limite le domaine d'application à des problèmes susceptibles d'être analysés complètement.

Les chercheurs en intelligence artificielle sont confrontés quotidiennement à des problèmes vastes et trop complexes pour être entièrement compris. Les humains résolvent ces derniers en utilisant l'heuristique (c'est-à-dire des règles empiriques) qui leur permet de réduire la taille du problème.

De par leur nature, les heuristiques peuvent conduire à des erreurs. En effet, elles ne garantissent pas toujours de trouver la réponse correcte, mais augmentent la probabilité de trouver une réponse susceptible d'être exploitée. L'heuristique dépend de la connaissance inhérente à une situation particulière et est souvent acquise par expérience.

Les techniques de programmation heuristiques vont rapidement accroître la variété des tâches pouvant être effectuées par les ordinateurs. Finalement, les techniques heuristiques vont dépendre des facteurs de confiance qui permettent aux experts humains d'évaluer leurs jugements.

Dès lors, les programmes heuristiques n'atteindront pas une réponse correcte, mais suggéreront différentes options de solution avec une probabilité associée à chacune.

1.2.3. Séparation des connaissances d'inférence et de contrôle

En programmation "traditionnelle", la connaissance se rapportant à un problème et les procédures pour manipuler cette connaissance sont mélangées. Ceci signifie qu'un expert humain doit dépendre d'un programmeur pour exprimer cette connaissance correctement.

Les chercheurs en intelligence artificielle ont développé un certain nombre de techniques pour séparer la connaissance d'un programme des procédures qui manipulent cette dernière. En effet, n'importe quel expert peut, par exemple, examiner la connaissance d'un système expert et déterminer si celle-ci est correcte ou non. De plus, quand la connaissance d'un problème donné change, l'expert peut situer précisément la ou les règles à modifier.

L'effet de cette nouvelle approche est de rendre la programmation accessible à "tout le monde". Etant donné que ces techniques d'intelligence artificielle deviennent mieux comprises, de nouveaux software sont conçus pour aider à la réalisation de différentes tâches spécialisées.

1.3. Conclusion

On peut évidemment discuter à l'infini sur la notion d'intelligence artificielle. Pour certains, l'ordinateur ne pourra jamais égaler le cerveau humain, car il n'accédera jamais au "sens" des informations qu'il traite. Mais pour d'autres, un ordinateur est déjà "intelligent" lorsqu'il est capable

d'accomplir certains travaux sans qu'on lui ait fourni l'algorithme nécessaire, c'est-à-dire une méthode rigoureuse définie en vue du résultat à atteindre, comme c'est le cas, par exemple, pour les opérations arithmétiques.

2. Introduction aux systèmes experts

2.1. Approche des systèmes experts

Fleurons de l'intelligence artificielle, les systèmes experts connaissent un développement croissant depuis le début des années 1980. Ce succès provient en partie du fait qu'ils ont permis d'aborder, avec des résultats significatifs, des tâches auparavant très peu informatisées relevant du diagnostic, de la prise de décision, etc. Il en a résulté un attrait croissant de l'industrie pour ces systèmes et pour les techniques d'intelligence artificielle en général.

L.Gailly-Goffaux et D.Ribbens (Réalisation d'un système expert) résument les raisons qui expliquent ce nouvel intérêt des entreprises en cinq points :

- besoin de nouvelles approches dans l'organisation du travail pour augmenter la productivité et faire face à la concurrence ;
- besoin d'expertise. Dans la majorité des disciplines, il y a peu de réels experts. De plus, il faut compter 10 à 15 ans pour qu'un être humain maîtrise une spécialité. Or, étant donné la tendance marquée des entreprises à la spécialisation, on assiste à une demande croissante de conseils d'experts ;
- besoin de connaissances. Le processus de prise de décision devient de plus en plus complexe et la nécessité d'une plus grande connaissance semble se faire sentir ;
- besoin de compétence. Les produits deviennent de plus en plus nombreux et complexes. Il est difficile de trouver suffisamment de personnes compétentes. Fournir des services compétents et appropriés semble devenir une véritable gageure ;
- besoin d'équipements automatisés plus "intelligents".

La plupart des systèmes véritablement experts en sont encore au stade des tests, de la maquette avancée ou du prototype expérimental. Ce qui ne veut pas dire qu'ils ne deviendront pas opérationnels un jour, mais nombre d'entre eux ne dépasseront pas le stade du prototype, faute d'avoir résolu leurs problèmes de généralisation, de temps de réponse, d'acceptation par les utilisateurs, d'ergonomie, etc. Le nombre de systèmes véritablement experts et réellement opérationnels ne doit guère dépasser aujourd'hui la centaine aux Etats-Unis.

2.2. Définition d'un système expert et composants

Les systèmes experts sont des programmes qui possèdent une grande quantité de connaissances dans un domaine spécialisé et qui sont utilisés pour résoudre des problèmes qui, de l'avis général, sont complexes et requièrent de l'expertise. Ils doivent être capables d'atteindre les performances des experts dans ce domaine ou, tout au moins, de fonctionner comme des assistants intelligents dans le cas où la décision finale est laissée aux experts humains.

Mettre en évidence la connaissance des experts, la modéliser et créer un système capable d'exploiter ce modèle de connaissances pour générer des raisonnements constituent les problèmes essentiels de la construction d'un système expert. De plus, de par la nature même de l'expertise humaine, à savoir un ensemble de méthodes informelles faisant intervenir l'expérience et l'intuition, il est quasiment impossible d'obtenir en une seule fois toutes les règles du raisonnement d'un expert humain. Il est admis en général que les experts ne peuvent décrire de manière explicite que 30 à 40% de ce qu'ils font et donc des règles qu'ils utilisent.

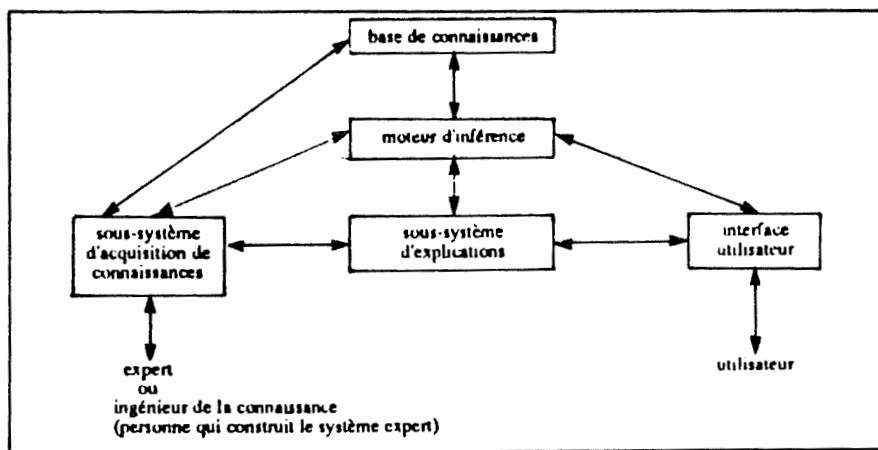


Figure 19 :
Architecture d'un système expert. [Gailly-Goffaux, 1988]

A présent, attachons-nous à l'architecture d'un système expert (voir figure 19).

Un système expert possède généralement :

- une ou plusieurs bases de connaissances constituant la mémoire à long terme du système, dans laquelle est codé l'ensemble des connaissances d'un domaine. Les règles de production sont largement utilisées pour représenter la totalité ou une partie des connaissances d'un expert. Il s'agit de parcelles élémentaires du type "SI <condition(s)> ALORS <conclusion(s)>", permettant de coder assez naturellement une démarche d'expert,
- une base de faits contenant des faits ou données propres à un problème à résoudre. Il s'agit donc d'une mémoire de travail qui s'enrichira de faits nouveaux découverts par le mécanisme de raisonnement jusqu'à ce que l'on parvienne à la solution du problème,
- un moteur d'inférences, algorithme chargé d'exploiter de façon cyclique les connaissances et les faits pour mener un raisonnement. Un moteur possède une certaine stratégie de fonctionnement,
- le sous-système d'explications, qui permet au système expert de fournir une explication, intelligible pour l'utilisateur, du raisonnement qu'il a suivi et donc de justifier ses conclusions. Ces explications sont nécessaires pour convaincre l'utilisateur du caractère correct de la solution fournie et de son adéquation au problème posé. Il permet également au développeur du système expert de s'assurer plus facilement que le système suit bien la démarche prévue,
- l'interface utilisateur, qui permet à l'utilisateur de dialoguer "naturellement" avec le système.

2.3. La connaissance et le raisonnement d'un système expert

Les systèmes experts résolvent des problèmes difficiles qui requièrent de l'expertise. Il s'agit d'un ensemble de connaissances approfondies d'un domaine particulier, où interviennent la compréhension des problèmes qui s'y posent et les capacités de résoudre certains de ceux-ci.

On peut distinguer trois types de connaissances dans un système expert :

- les "faits" constituent un ensemble de connaissances explicites, publiquement disponibles et reconnues par les experts d'un domaine,
- les "règles" permettent de déduire des connaissances implicites à partir des faits,
- les "heuristiques" consistent en des "règles de bonne pratique" acquises grâce à l'expérience.

Le raisonnement d'un système expert s'effectue généralement dans un univers de connaissances incomplet et incertain. Contrairement aux applications algorithmiques "traditionnelles", la plupart des systèmes experts travaillent dans des situations qui n'admettent pas de solutions optimales. Dès lors, le système doit trouver le meilleur compromis possible entre la qualité de la réponse qu'il fournit et l'effort nécessaire à cette réponse. Un principe important dans l'organisation d'un système expert est de maintenir un raisonnement qui soit compréhensible pour le spécialiste du domaine.

2.4. L'explication du raisonnement

Pour être acceptable, un système expert doit posséder la capacité d'expliquer son raisonnement.

Ceci est important pour différentes raisons, parmi lesquelles :

- la compréhension du raisonnement : le concepteur du système et l'utilisateur ont besoin de comprendre le contenu de la base de connaissances et les étapes du raisonnement afin de mettre à jour et d'utiliser efficacement le programme,
- la mise au point du système expert : le processus d'explication permet au programmeur et à l'expert de vérifier si tous les aspects du problème ont été pris en considération,
- la persuasion de l'exactitude des résultats fournis par le système expert : il s'agit de convaincre les utilisateurs que les conclusions du système sont rationnelles.

2.5. Types de systèmes experts

Le but de ce paragraphe est de donner une idée des applications possibles pour les systèmes experts. Ces derniers peuvent être classifiés selon la tâche générique pour laquelle ils sont construits. Parmi les différentes tâches possibles, citons :

2.5.1. L'interprétation

Le problème est d'analyser des données pour déterminer leur signification. Ces systèmes peuvent être utilisés pour la compréhension de la parole, l'analyse d'images, l'élucidation de structures chimiques, l'interprétation de signaux,...

2.5.2. La prédiction

L'objectif est de prévoir le comportement du futur à partir d'un modèle du passé et du présent. Le système raisonne sur le temps, en ce sens qu'il fait référence à des données qui changent dans le temps et à des événements ordonnés dans le temps. Un tel système peut être utilisé pour des prévisions météorologiques, démographiques, militaires,...

2.5.3. Le diagnostic

Ces systèmes analysent des données en se basant sur l'interprétation d'observations. Les diagnostics sont utilisés dans de nombreux domaines tels que la médecine, l'électronique, la mécanique, le logiciel,...

2.5.4. La planification

Il s'agit de préparer un programme d'actions pour réaliser des buts donnés sans consommer des ressources excessives, ni violer des contraintes données. La planification est utilisée dans des domaines tels que la programmation automatique, la robotique, l'élaboration de plans militaires,...

2.5.5. Le monitoring

Le problème est d'interpréter de manière continue des signaux et de déclencher une alarme quand une intervention est requise. La médecine utilise de tels systèmes pour la surveillance des malades et la correction des troubles.

2.5.6. Le contrôle

Le contrôle est une des tâches les plus compliquées pour un système expert. Cette tâche est multiple : il s'agit d'interpréter, prédire, détecter les vulnérabilités, développer et appliquer des remèdes aux dysfonctions en observant le comportement des systèmes, diagnostiquer les causes de problèmes anticipés,...

Un tel système peut, par exemple, être utilisé pour le contrôle du trafic aérien.

2.5.7. L'instruction

Un système expert d'instruction consiste en un système intelligent d'enseignement assisté par ordinateur. Un tel système est plus puissant que les systèmes EAO traditionnels, et ce à cause de la flexibilité d'apprentissage permise.

Les ICAI (Intelligent Computer-Assisted Instruction) ont une architecture organisée selon quatre types de connaissances :

- la connaissance du domaine d'application,
- la compréhension que possède l'étudiant dans ce domaine,
- la connaissance relative à l'apprentissage (comment apprendre ?),
- la connaissance liée à la communication (comment communiquer avec l'élève ?).

Chacun de ces types de connaissances constitue un module fonctionnel distinct dans le système ICAI.

L'objectif d'un tel système d'enseignement intelligent assisté par ordinateur est d'accroître les progrès de l'étudiant mais surtout d'encourager l'apprentissage.

Nous citerons ci-dessous quelques ICAI parmi les plus populaires ; pour plus de renseignements, le lecteur consultera le chapitre 9 du volume 2 du Handbook of Artificial Intelligence: Applications-oriented AI research : education.

- WHY, décrit par Stevens et Collins (1977) guide les étudiants dans les causes des pluies annuelles, processus géophysiques complexe qui dépend de nombreux facteurs interreliés.
- GUIDON, programme d'application aux problèmes de diagnostic des maladies a été développé par W.J.Clancey et ses collègues de l'Université de Stanford. En utilisant les règles de MYCIN, GUIDON invite l'étudiant à un dialogue au sujet d'un patient suspecté d'avoir contracté une maladie infectieuse.

2.5.8. La mise au point

L'objectif est d'éliminer les erreurs en prescrivant des remèdes pour les dysfonctions. La mise au point est un domaine encore à l'état de recherche. Elle est très utile, par exemple, pour la conception de programmes informatiques.

2.6. Choix du domaine d'application d'un système expert

Il n'est pas simple de décrire les caractéristiques qui rendent un problème approprié au développement d'un système expert. La première tâche des ingénieurs de la connaissance sera de s'assurer que le projet est réalisable.

Pour évaluer la "faisabilité" d'une application possible, il est intéressant de disposer d'un ensemble de références d'attributs que doit posséder un "bon domaine pour

système expert". Nous allons présenter quelques-uns de ces attributs. Cependant, il faut remarquer, d'une part, qu'il est peu probable qu'un domaine puisse vérifier tous les attributs et, d'autre part, l'importance relative de chacun de ceux-ci.

2.6.1. Caractéristiques de base

- le domaine est caractérisé par l'usage des connaissances, du jugement et de l'expérience d'experts ;
- à l'heure actuelle, il y a des experts reconnus qui résolvent le problème ;
- l'expertise n'est pas disponible de manière permanente et fiable ; il y a donc un besoin de "capturer" cette expertise.

2.6.2. Type de problème

- la tâche requiert essentiellement un raisonnement symbolique et l'utilisation d'heuristiques. Elle peut, en outre, nécessiter l'examen d'un très grand nombre de possibilités ou une prise de décision basée sur des informations incomplètes ou incertaines ;
- la tâche ne requiert pas de connaissances provenant d'un très grand nombre de domaines diversifiés ;
- la tâche est définie très clairement : dès le début du projet, il devrait y avoir une définition précise des données et des résultats du système expert à développer.

2.6.3. L'expert

- il existe un expert qui pourra travailler sur le projet ;
- l'expert pourra consacrer une partie importante de son temps au développement du système ;
- la connaissance et la réputation de l'expert sont telles que, si le système expert "capture" une partie de l'expertise de cet expert, les solutions proposées par le système pourront avoir de la crédibilité et de l'autorité ;
- l'expert est capable de communiquer ses connaissances et son expérience ainsi que les méthodes qu'il utilise pour résoudre les problèmes.

2.6.4. Autres caractéristiques souhaitables

- la tâche doit être décomposable pour permettre la mise au point rapide d'un prototype pour un sous-ensemble de cette tâche et ensuite une extension progressive jusqu'à la tâche complète ;
- la tâche n'est pas du type "tout ou rien" : un pourcentage de résultats incorrects ou non-optimaux peut être toléré ;
- il existe de nombreux cas de tests, des exemples utilisables ;
- le domaine est relativement stable. Les changements pourront amener à une révision de la base de connaissances, mais pas à une modification totale du mode de raisonnement ;
- le personnel du domaine (c'est-à-dire les futurs utilisateurs du système expert) et les experts acceptent l'idée d'un tel système, acceptent de participer à sa construction et comprennent que le système est limité en portée et, comme un expert humain, ne peut pas produire des résultats corrects ou optimaux dans 100% des cas. Cette acceptation de l'idée du système expert est en fait fort importante pour le bon déroulement du travail de longue haleine qu'est la construction d'un tel système.

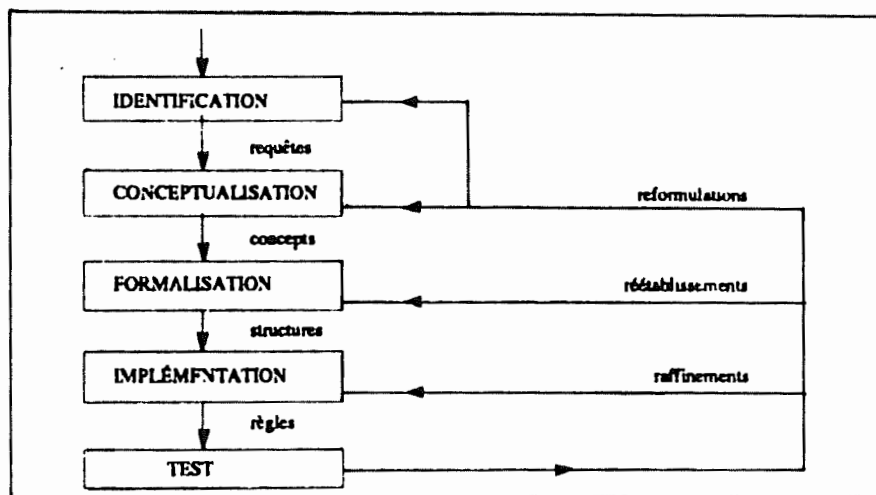


Figure 20 :
Etapes intervenant dans la construction d'un système expert.
[Gailly-Goffaux, 1988]

La sélection du domaine d'application du système expert se fera au cours de discussions entre quelques experts du domaine, quelques utilisateurs potentiels et l'équipe d'intelligence artificielle.

Puisque nous venons de parler de l'équipe d'intelligence artificielle, nous pouvons aborder maintenant le problème de sa composition. On estime généralement que cette équipe doit comprendre un ingénieur de connaissances (parfois appelé cogniticien) et un programmeur de haut niveau. L'ingénieur de connaissances "extraît" des experts humains leurs connaissances pour résoudre les problèmes du domaine. Ce processus d'acquisition des connaissances consiste au transfert et à la transformation de l'expertise pour la résolution d'un problème depuis une source de connaissances jusqu'à un programme. Le programmeur de haut niveau s'attachera essentiellement aux aspects plus techniques liés à l'implémentation du système.

2.7. Une méthode de construction d'un système expert

Bien qu'il n'existe pas encore de méthode de construction de systèmes experts qui soit universellement acceptée, les spécialistes du domaine s'accordent généralement sur une succession d'étapes que nous allons brièvement décrire.

Un système expert étant construit selon un processus de développement itératif, l'acquisition des connaissances n'est pas directe et définitive. Il en résulte que la démarche présentée ne suit pas toujours un ordre chronologique strict mais qu'il existe des recouvrements possibles. La figure 20 résume les différentes étapes nécessaires à la modélisation de la connaissance.

2.7.1. L'étape d'identification

La première étape dans le processus d'acquisition des connaissances consiste à dégager les éléments essentiels intervenant dans le problème, et à rechercher les relations entre ces éléments, les classifications et les hiérarchies éventuelles.

L'identification ne peut se faire qu'en entretenant des rapports réguliers (au moins hebdomadaires) avec les experts du domaine.

Classiquement, il s'agit :

- de déterminer les participants au développement,
- de définir les caractéristiques du problème,
- d'évaluer les ressources nécessaires,
- de déterminer les objectifs du système.

a. Identification des participants et de leur rôle

L'équipe de développement du système expert peut être composée d'experts du domaine, d'ingénieurs de connaissances et de spécialistes interdisciplinaires.

Dans les premiers mois de développement, il est conseillé de recourir à un petit nombre d'experts du domaine afin d'établir un consensus. Par la suite, il sera intéressant de soumettre les connaissances déjà acquises à d'autres experts, de les confronter au résultat de ce regroupement afin de préciser et d'affiner la connaissance.

L'expert doit être une personne qui connaît le domaine depuis plusieurs années. Il ne lui suffit pas d'avoir une compréhension théorique du domaine ; il doit également avoir une grande expérience pratique.

Puisque les ingénieurs de connaissances ont souvent peu de connaissances du domaine, leur rôle se limite, dans un premier temps, à se familiariser avec le domaine et les problèmes qu'on y rencontre. Les ingénieurs de connaissances doivent être capables d'assimiler une grande partie des connaissances de base du domaine. Leur rôle est de "pousser" les experts à fournir leur connaissance et à formuler explicitement les mécanismes qu'ils utilisent pour résoudre leurs problèmes.

Il faut aussi essayer de leur "extraire" la connaissance qu'ils ont accumulée durant leur expérience. Ceci n'est pas toujours évident : en effet, les experts du domaine ont en général de grandes difficultés à exprimer leurs savoirs et le raisonnement qu'ils suivent pour résoudre un problème. De plus, ils se basent quelquefois sur l'intuition.

b. Identification du problème

Lorsque les participants ont été choisis, l'ingénieur de connaissances et l'expert peuvent procéder à l'identification du problème. L'objectif est double : il s'agit d'une part de cerner le problème, et d'autre part d'identifier les structures de connaissances pertinentes à sa résolution.

Voici quelques questions qui sont fréquemment abordées :

- Quelle classe de problèmes le système expert devra-t-il résoudre ?
- Quels sont les sous-problèmes importants ?
- Quelles sont les données ?
- Quels sont les concepts clés et leurs relations ?

- A quoi ressemble une solution et quels sont les concepts qu'elle fait intervenir ?

Il est courant de devoir envisager l'identification du problème à plusieurs reprises.

c. Identification des ressources

Des ressources typiques sont les sources de connaissances, le temps, les moyens informatiques et l'argent. Elles sont nécessaires pour acquérir la connaissance, l'implémenter et la tester.

L'expert du domaine puisera des connaissances dans son expérience professionnelle, la littérature du domaine, des études de cas et des bases de données.

L'ingénieur de connaissances se référera à des problèmes analogues traités dans le domaine des systèmes experts. Il y découvrira des méthodes, des représentations et des outils propres à ce domaine.

Le temps est une ressource critique. L'ingénieur de connaissances et l'expert doivent consacrer plusieurs mois d'activité pour permettre la construction d'un premier prototype.

Enfin, les moyens informatiques (logiciels et matériels) doivent rester disponibles tout au long du processus de développement.

d. Identification des objectifs

Décrire les objectifs du système, c'est décrire l'utilité du système dans l'endroit où il sera implanté. Quelques exemples d'objectifs sont la distribution d'une expertise rare, l'amélioration des méthodes de résolution de l'expert, ou encore l'automatisation des aspects routiniers du travail de l'expert,...

2.7.2. L'étape de conceptualisation

Cette seconde étape a pour but d'extraire des discussions un ensemble de faits, d'heuristiques, de raisonnements pour résoudre les problèmes qui sont posés à l'expert. Il s'agit également de mettre en évidence des corrélations entre ces faits et raisonnements : les grouper, les généraliser.

Cette étape doit prendre la forme d'un processus interactif entre les participants au développement. Lors des entretiens, différents moyens peuvent être utilisés : dessins, graphiques, organigrammes, etc. L'ingénieur peut commencer à décrire une architecture globale pour la base de connaissances et les mécanismes de contrôle qui viendront se greffer sur le raisonnement de l'expert.

2.7.3. L'étape de formalisation

Cette troisième étape a pour objectif de déterminer un langage approprié pour les concepts relevés. Pour ce faire, l'ingénieur de connaissances étudie la structure des concepts, leurs interrelations, la nature des données, etc.

Le résultat sera un ensemble de spécifications partielles décrivant comment le problème peut être représenté avec les outils choisis. Elles peuvent servir pour la construction de la base de connaissances d'un prototype du système.

2.7.4. L'étape d'implémentation

Il s'agit de combiner et d'organiser la connaissance formalisée à l'étape précédente afin qu'elle rencontre les spécifications requises par le moteur d'inférence qui devra l'exploiter. Ce travail rigoureux aide à déceler les lacunes et incohérences de la base de connaissances développée.

Le résultat de cette étape est un programme exécutable, prototype du système expert.

2.7.5. L'étape de test

Cette cinquième étape va s'attacher à valider la base de connaissances que l'on a établie. Il s'agit de tester le prototype sur des exemples de complexité croissante afin de déceler les lacunes de la base de connaissances et du moteur d'inférence utilisé.

Le fait d'utiliser des exemples de complexité croissante et de les "faire tourner" successivement permet d'isoler plus rapidement et plus aisément les problèmes éventuels que le traitement de ces exemples fait apparaître. L'expert évalue les performances sur chaque exemple et assiste l'ingénieur de connaissances dans ses révisions.

Ces révisions peuvent impliquer une reformulation des concepts, un rétablissement des structures de représentation ou encore un raffinement du système implémenté. Ces tâches imposent un cyclage à travers les étapes jusqu'à ce que le comportement attendu soit obtenu.

On peut également penser, à ce stade, à évaluer l'utilité du système. On se demande si la solution au problème, telle qu'elle est proposée par le système, aide l'utilisateur de manière significative.

2.7.6. La mise au point du prototype

Le but de cette étape est de produire un système expert véritable, c'est-à-dire ayant des performances comparables à celles d'un expert humain. Ainsi, l'ingénieur de connaissances fera évoluer progressivement la base de connaissances de son système. Cette évolution peut nécessiter une modification profonde de l'organisation et de la représentation initiales de la connaissance.

Par ailleurs, on ne peut véritablement parler de système expert que si celui-ci possède un certain nombre de propriétés rendant son utilisation agréable et efficace.

Parmi les extensions utiles pour un tel système, l'ingénieur de la connaissance devra considérer par exemple :

- une interface utilisateur facile à utiliser, l'idéal étant bien sûr une interface en langage naturel,
- différentes possibilités d'examen de la base de connaissances et du raisonnement suivi,
- une librairie des cas présentés au système,
- des possibilités de réponse aux différents utilisateurs quand le constructeur est absent,
- etc.

Nous pouvons conclure en disant que le processus complet d'acquisition de connaissances est un processus incrémental procédant par affinements successifs et cyclant à travers les étapes décrites ci-dessus.

2.8. Evaluation d'un système expert

Pour rédiger cette section, nous nous sommes basées sur l'article de R.M.O'Keefe, O.Balcio et E.P.Smith : Validating Expert System Performance.

L'étape de test du système expert est continue et se fait au fur et à mesure du processus incrémental d'élaboration du système. Toutefois, le système terminé, on souhaiterait pouvoir l'évaluer. Les experts et l'équipe d'intelligence artificielle devraient pouvoir répondre à des questions de ce type :

- le schéma de représentation de la connaissance est-il adéquat ?
- le système fournit-il des réponses correctes avec de bonnes justifications ?
- l'interface avec l'utilisateur est-elle satisfaisante ?
- le système est-il efficace, rapide et fiable ?

On procède généralement par comparaison des résultats fournis par le système, d'une part, et par des experts humains dans des cas réels d'expertise, d'autre part. Les ingénieurs de connaissances calculent un pourcentage de réussite du système et utilisent un jugement subjectif pour analyser et expliquer les échecs du système expert, c'est-à-dire les endroits où les résultats du test contredisent des résultats connus ou l'opinion des experts.

Ainsi, il existe des systèmes ayant de bonnes performances. Ne citons que PUFF, un système expert qui interprète un ensemble de résultats de tests respiratoires et produit un diagnostic pulmonaire du patient ; ce système, testé sur plus de 4000 cas, fournit des rapports corrects dans 85% des cas.

Cependant, cette approche simple présente quelques problèmes. Le pourcentage final obtenu dépend du choix des jeux de tests, et son exactitude est fonction du nombre de jeux de tests choisis.

Dès lors, nous allons essayer d'établir une validation formelle du système expert et tenter de présenter des méthodes qualitatives de validation.

Avant d'aller plus loin dans ce développement, il nous semble important de préciser quelques concepts.

2.8.1. Vérification, validation et évaluation

Typiquement, la validation est une partie de l'évaluation. Cette dernière cherche à affecter une valeur générale au système expert, tandis que la validation se réfère, elle, à la construction d'un "bon" système, c'est-à-dire un système qui s'exécute avec un degré d'exactitude acceptable. Le concept de vérification s'attache à la construction d'un système "juste", c'est-à-dire qu'il prouve que le système implémente correctement ses spécifications.

Nous examinerons successivement les problèmes rencontrés dans la validation de l'exécution d'un système expert, aborderons les concepts fondamentaux de la validation et, finalement, retiendrons quelques méthodes qualitatives appropriées.

2.8.2. La validation des systèmes experts

Les problèmes majeurs qui se posent sont :

- QUE valider ?
- valider CONTRE QUOI ?
- valider AVEC QUOI ?
- QUAND valider ?
- COMMENT CONTROLER le coût de validation ?
- COMMENT TRAITER des résultats multiples ?

a. Valider quoi ?

Nous pouvons valider les résultats intermédiaires, le résultat final (souvent appelé la conclusion), le raisonnement du système ou toute combinaison de ces trois. Ce qu'il faut valider est intrinsèquement lié à l'étape de développement atteinte. En effet, à toute étape, si une partie de l'exécution du système peut être mesurée pour un ensemble de données d'"entrée", cette partie devrait être validée de telle sorte que les erreurs soient repérées le plus tôt possible dans le cycle de développement.

b. Valider contre quoi ?

Les systèmes experts peuvent être validés contre des résultats aussi bien que contre les performances des experts du domaine.

c. Valider avec quoi ?

Dans un monde idéal, de nombreux cas de tests (documentés par des experts) devraient être disponibles pour valider le système. Malheureusement, dans le monde réel, très peu de tests sont disponibles pour effectuer ce travail.

Pour une bonne validation, des cas de tests devraient être sélectionnés aléatoirement en utilisant un échantillonnage dans chaque classe de résultats identifiable. Par exemple, si un expert classifie un diagnostic comme A, B ou C - et on sait que ces classifications se retrouvent respectivement 80%, 15% et 5% du temps - alors, une collection de 200 cas de tests incluerait 160, 30 et 10 instances où A, B et C résulteraient respectivement.

d. Quand valider ?

Les systèmes experts doivent montrer des performances acceptables dès le début du développement. Mais le niveau d'acceptation peut être différent selon les systèmes : avec un prototype de recherche, une performance moyenne peut être acceptable, moyennant des indications montrant que l'approche de base est correcte. Quand le système s'étend, une validation supplémentaire peut s'avérer nécessaire pour améliorer ou affiner le système. Dans des applications critiques où des vies humaines sont en jeu ou des fortunes en danger, tester dans le domaine d'application est souvent impossible.

e. Comment contrôler le coût de la validation ?

La validation, souvent consommatrice de temps, est chère. Fixer avec précision le montant d'argent et de temps qui devrait y être consacré est une tâche difficile voire impossible. Il est admis que les coûts de validation peuvent être contrôlés en concevant des méthodes formelles de validation qui soient intégrées dans le processus de construction du système expert.

f. Comment traiter les résultats multiples ?

Quand on valide un système expert à résultats multiples, on se trouve confronté à un sérieux obstacle. En effet, nous ne pouvons pas tester la validité d'une réponse multiple en testant séparément chacune de ses composantes.

Dans un système expert de diagnostic médical dont l'objectif est de prescrire le traitement adéquat, deux médicaments peuvent être prescrits et chacun d'eux peut constituer un traitement à lui seul, mais la combinaison des deux peut s'avérer inacceptable dans certains cas. Ainsi, la réponse globale du système est inadéquate.

2.8.3. Quelques concepts fondamentaux de la validation

Notons tout d'abord qu'un système expert n'est jamais entièrement valide ou invalide. En effet, étant donné que ces systèmes sont des représentations des abstractions de la réalité, nous ne pouvons pas nous attendre à un comportement parfait.

Les concepteurs du système vont travailler dans le cadre d'une application bien précise et, dès lors, ce système peut être valide dans ce domaine et complètement absurde dans un autre.

Nous pouvons également distinguer deux niveaux dans la formalisation de la validation.

La validation formelle établit quand la validation devrait avoir lieu dans le cycle de développement, et identifie des méthodes de validation, le domaine de spécification, le niveau d'acceptation et la possibilité d'appliquer des techniques statistiques.

La validation informelle - typiquement réalisée à la fin du développement - emploie des méthodes ad hoc et est souvent considérée comme un réflexe d'après coup.

2.8.4. La validation qualitative

Cette méthode de validation emploie des comparaisons subjectives de performances. Nous avons relevé quelques approches de validation qualitative parmi lesquelles certaines ont déjà fait leurs preuves.

a. Validation de "surface"

Les membres de l'équipe du projet, les utilisateurs potentiels du système et les experts du domaine d'application comparent subjectivement les résultats du système expert avec ceux des experts humains.

b. Tests de Turing

Le mathématicien A.Turing avait émis un critère d'évaluation global d'un système expert : on place derrière une tenture un ordinateur (avec le système expert) et des experts. Si l'utilisateur ne peut distinguer qui du système ou des experts lui fournit la réponse à sa question, le système expert a brillamment passé l'épreuve. De tels tests ont été utilisés pour valider MYCIN.

c. Tests dans le domaine d'application

Ces tests placent le système expert dans "son" environnement de travail, et ont pour but de repérer principalement les erreurs d'exécution. Cette technique possède l'avantage de placer le poids du test sur les utilisateurs.

d. Validation des sous-systèmes

Cette méthode suppose que le système puisse être divisé en un certain nombre de sous-systèmes permettant d'observer les performances individuelles de chacun de ceux-ci.

e. Interaction visuelle

Il s'agit d'une animation du système expert permettant aux experts d'interagir, de modifier des paramètres, et ce autant de fois qu'ils le désirent.

Il est à noter qu'une validation quantitative existe également ; cette méthode, principalement basée sur des techniques statistiques, n'a offert que peu de résultats tangibles, et nous ne la développerons pas ici.

Il apparaît clairement que les notions de validation qui viennent de vous être présentées sont pour la plupart fort descriptives. Cependant, les concepteurs de systèmes experts ont besoin d'une méthodologie établie, c'est-à-dire une méthode expliquant comment valider ces systèmes sous certaines conditions et contraintes. De nos jours, l'expérience en termes de validation des systèmes est très limitée. Les méthodologies n'évolueront que dans la lumière d'une future expérience collective et d'une appréciation critique de cette expérience.

2.9. Conclusions

La réalisation d'un véritable système expert nécessite un important investissement tant en personnel, qu'en temps et en argent. Elle n'est pas aisée : on ne dispose pas actuellement d'une méthodologie bien définie de construction de systèmes experts ni de librairies de programmes. Elle requiert des machines puissantes, des informaticiens très spécialisés, une grande disponibilité des experts et pour bien faire leur enthousiasme. On peut cependant se poser les questions suivantes : L'expert existe-t-il ? Est-il prêt à transmettre sa connaissance ? Dans bien des cas les réponses sont négatives. Les systèmes experts éveillent beaucoup de suspicion et nombre de responsables refusent de mettre leurs meilleurs experts à la disposition de l'ingénieur de connaissances. Si, par chance, l'expert devient accessible, il refuse souvent de coopérer activement car, son savoir ou son savoir-faire accumulé avec les années constitue la substance-même de l'intérêt qu'il présente pour la société. Le céder, à une machine de surcroît, c'est aller à l'encontre de la culture des entreprises où, pour avancer, il faut savoir se rendre indispensable.

Cependant, les systèmes experts présentent encore d'importantes limitations :

- la nécessité de construire "à la main" l'ensemble de la base de connaissances et l'absence quasi-totale d'apprentissage automatique : un système expert ne s'améliore pas pour l'instant avec l'expérience,
- le manque de connaissances du système expert sur ses propres limites de compétence,
- l'aspect superficiel des connaissances exploitées par le système expert, par opposition à la connaissance profonde des phénomènes sous-jacents, sur laquelle l'expert humain appuie son raisonnement,
- la limitation des modes de raisonnement mis en oeuvre.

Toutefois, l'intérêt de la construction du système expert peut être à la hauteur de l'investissement consenti. Le système permet d'abord de disposer d'une modélisation de la connaissance dans un domaine particulier alors que jusque-là cette formalisation n'existait pas. Le système permet de conserver l'expertise après le départ de l'expert (retraite,...). Il peut également être utile pour former de nouvelles personnes. Dans une autre optique, un système expert peut être rapidement rentabilisé car, s'il est bien accepté, il constitue un expert supplémentaire. Cet expert n'aura pas de défaillances ni d'oublis : il envisagera toujours toutes les possibilités, même les cas peu fréquents que l'on pourrait avoir perdu de vue. Dans les cas où il est difficile d'avoir un expert à sa disposition (manque d'experts, éloignement géographique,...), un système expert peut s'avérer fort utile. Pour tout cela, il ne faut pas perdre de vue que le système expert, avec ses connaissances dans un domaine très étroit et ses échecs (comme un expert humain) doit être réellement accepté dans l'entreprise.

On peut penser, en spéculant sur le futur, que l'on disposera de meilleurs outils de construction de systèmes experts. On peut par exemple imaginer que l'on aura des librairies de procédures pour le diagnostic, le design, la planification,..., des librairies de faits, des bases de connaissances ne démarrant pas à partir de rien. On pourrait aussi disposer de meilleurs systèmes d'explications et de debugging, de nouveaux modes d'interfaçage comme la voix ou la vision. On aura peut-être des modèles des utilisateurs de telle sorte que les systèmes pourront s'adapter à la personne qui les consultera. En tout cas, on peut espérer que les ingénieurs de connaissances acquerront une plus grande expérience dans leur domaine (pourquoi ne pas parler d'expertise ?) et que les méthodologies se dégageront.

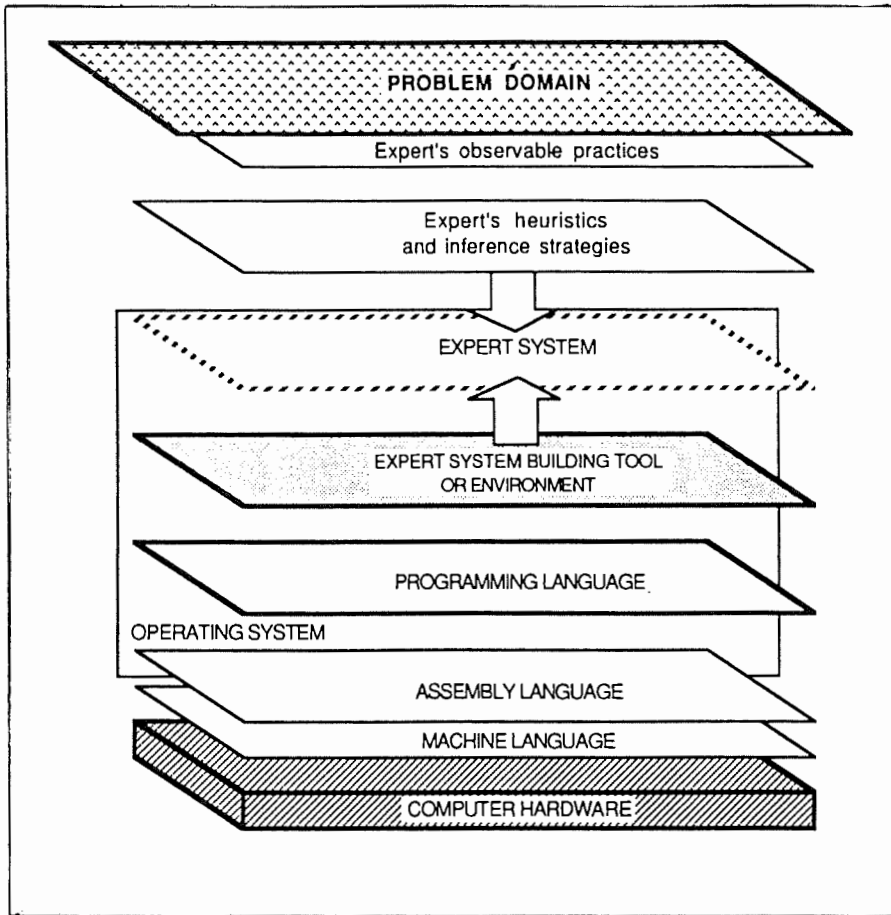


Figure 21 :
Niveaux de software s'échelonnant du niveau application à la machine (hardware).
[P.Harmon et al., 1988]

3. Langages et outils de construction de systèmes experts

La construction d'un système expert requiert bien évidemment l'utilisation d'un langage ou d'un outil de développement. Ces deux moyens sont de fait à distinguer, et nous allons nous employer à définir quelques caractéristiques permettant de les différencier.

Nous allons, dans un premier temps, nous attarder sur les langages de programmation ; ensuite, nous étudierons quelques outils de développement, et plus particulièrement VP-Expert - l'outil avec lequel nous avons implémenté notre prototype de système expert -, Nexpert et Guru.

La rédaction de cette section est basée sur les chapitres 3 et 4 de l'ouvrage de P.Harmon, R.Maus et W.Morrissey : Expert Systems Tools And Applications.

3.1. Langages

Dans la hiérarchie des langages, le plus bas niveau auquel l'homme, le programmeur opère généralement est l'assembleur. Toutefois, il n'est pas utile de préciser que ce langage est ... "ennuyeux", et qu'il est plus intéressant, et plus aisé de travailler avec des langages de plus haut niveau, tels que Basic, Cobol, Fortran, Pascal, PL/1, C, ou ADA (voir figure 21).

Certains de ces langages sont évidemment plus appropriés que d'autres pour telle tâche, ou tel contexte. Ainsi, Fortran est surtout employé dans la programmation de procédures ayant trait au monde scientifique, alors que le développement de programmes manipulant des données financières ou économiques se fait principalement à l'aide de Cobol.

Ajouté à cela, beaucoup d'autres langages de haut niveau sont utilisés pour développer des programmes dans des domaines spécifiques.

Par exemple, les deux langages auxquels on a le plus couramment recours en intelligence artificielle sont Lisp et Prolog. Notons que ces langages tournent mieux avec des "operating systems" écrits eux-mêmes dans un langage d'intelligence artificielle, ou sur des machines conçues à cet effet (comme les machines Lisp, par exemple).

A un niveau supérieur se situent les environnements de programmation, comme Lotus 1-2-3 et dBase III, destinés à faciliter l'implémentation d'une tâche particulière.

Dans un premier temps, nous allons décrire brièvement quelques paradigmes de programmation. Nous établirons ensuite une comparaison succincte entre les langages conventionnels et l'ingénierie de connaissances, avant d'exposer plus en détails certains langages, à savoir Lisp, Prolog, et les langages orientés objet.

3.1.1. Les paradigmes de programmation [Barreto]

Tout d'abord, il serait utile de préciser ce que l'on entend par paradigme. Un paradigme de programmation correspond à un style de programmation ; ceci implique alors la construction d'associations telles que :

- Prolog et programmation logique,
- Lisp et programmation fonctionnelle,
- Pascal et programmation impérative,
- Smalltalk et programmation orientée objet,
- etc...

a. Style déclaratif versus style impératif

Dans le style déclaratif, on décrit le résultat souhaité, et on fournit les informations nécessaires à la réalisation d'une tâche, en l'occurrence la résolution d'un problème. Aucune combinaison des informations disponibles n'est imposée ; le "comment" parvenir au résultat n'est donc pas explicite.

D'autre part, dans le style impératif, on trouve la description des opérations qui seront exécutées. Le résultat final est une conséquence de l'exécution de la suite des opérations ; il existe donc une certaine idée d'ordre.

Il est à noter qu'au sein d'un même programme, on retrouve souvent les deux styles de programmation. Ainsi Fortran et Pascal favorisent le style impératif (assignations, structures de contrôle, ...) ; toutefois, les déclarations de variables correspondent au style déclaratif.

Par ailleurs, Lisp est principalement basé sur le style fonctionnel ; en effet, souvent la définition d'une fonction se fait par récursivité. Cependant, pour des raisons pratiques liées à l'implémentation, cette définition peut parfois se faire de manière impérative.

De même pour Prolog qui au départ présente surtout le style déclaratif : une exécution correcte d'un programme dépend de l'ordre d'exécution des instructions, et certaines actions sont de nature impérative (l'ouverture d'un fichier, par exemple). N'oublions pas toutefois que ce langage est également basé sur un paradigme logique.

b. La programmation logique

Dans le domaine de démonstration de théorèmes, le but est de prouver que le théorème est vrai ou faux. En programmation logique - comme dans toute autre activité de programmation - l'objectif est de résoudre un problème. D'une certaine façon, la programmation logique peut être considérée sous cet angle, et c'est ce point de vue qui constitue la base de Prolog.

En fait, ce langage est apparu pour aider à la construction de programmes de démonstration de théorème. Il est cependant loin d'une programmation logique idéale, mais il est un exemple représentatif d'un langage favorisant ce paradigme.

c. La programmation fonctionnelle

Dans le paradigme de programmation fonctionnelle, le langage est considéré comme possédant un ensemble de fonctions (primitives), et un processus bien défini permettant de combiner ces fonctions, et d'en définir d'autres (composition de fonctions, récursivité).

Dans ce style, des éléments tels que 'goto', boucles, assignations sont exclus.

L'exemple le plus évident d'un langage présentant une telle tendance est Lisp.

Notons qu'ici aussi il est rare que seul le caractère fonctionnel apparaisse ; en effet, de nombreux mécanismes extérieurs à ce paradigme sont employés.

d. La programmation orientée objet

La caractéristique de base de ce style de programmation est l'objet ; celui-ci est une entité logique. Ces objets sont groupés en classes, une classe correspondant à un objet complexe structuré qui caractérise un ensemble d'objets, et qui présente les propriétés de ces objets.

Une instance d'une classe est un élément particulier de cette classe.

Smalltalk fut développé sur base du paradigme orienté objet. Plus récemment, d'autres langages présentant des propriétés d'héritage, de transfert de messages, ... - comme Loops, Xlisp, ... - ont été implémentés.

En guise de conclusion, nous pouvons ajouter tous ces styles de programmation peuvent "coexister", avec toutefois une tendance vers un paradigme particulier.

Ainsi C++ est surtout orienté vers un style impératif tout en ayant certaines parties déclaratives ; de plus, il est possible d'observer des mécanismes d'héritage, ce qui sous-tend un paradigme orienté objet.

3.1.2. Software symbolique et software conventionnel

[HARMON88]

Outre la classification effectuée ci-dessus, il est également possible de grouper les software en deux grandes classes : software symbolique et software conventionnel.

Comme nous l'avons déjà décrit ci-dessus, les principaux langages conventionnels utilisés sont Fortran, Cobol, Pascal et C, chacun étant tout de même spécialement employé pour des tâches ou des domaines différents.

Toutefois, ceux-ci partagent des principes de base : l'homme développe un algorithme pour résoudre un problème en procédant pas par pas, et les données nécessaires à une exécution correcte

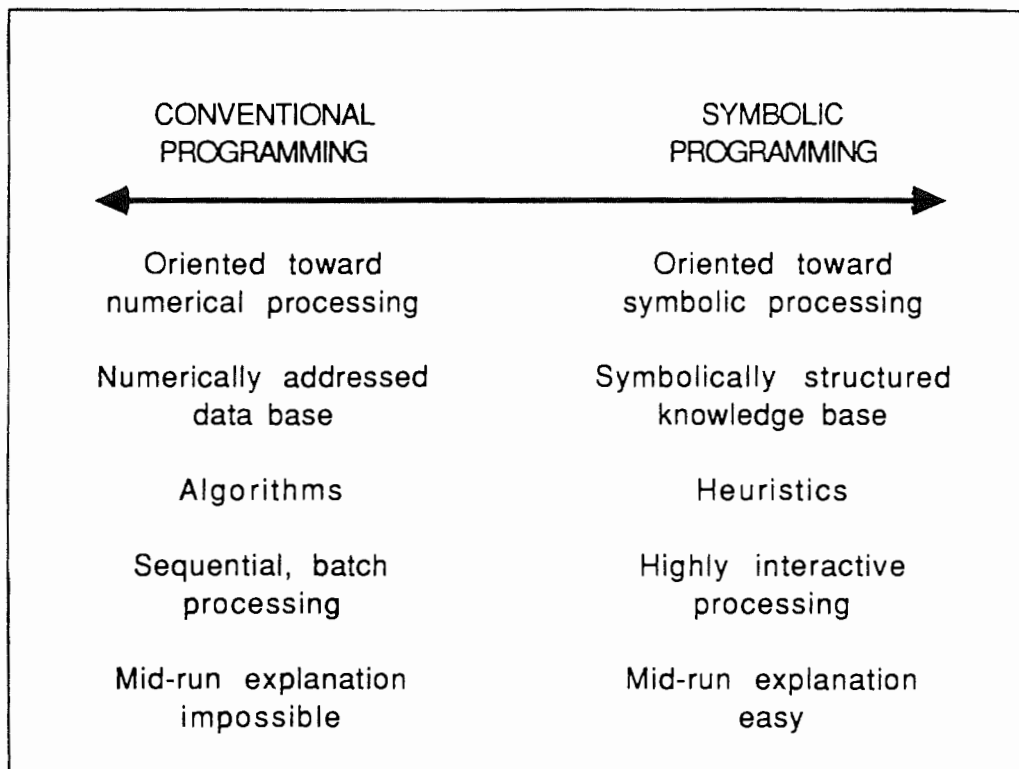


Figure 22 :
Différences entre les programmations conventionnelles et symboliques.
[P.Harmon et al., 1988]

sont stockées dans une base de données, et appelées au moment voulu. Afin de rendre le programme exécutable d'une façon optimale, le programmeur l'améliore petit à petit, de sorte qu'il devient de plus en plus difficile de savoir exactement et à tout moment ce que le programme réalise.

Au moment où Fortran prenait de l'ampleur, Mc Carthy, imminent chercheur en intelligence artificielle, développait Lisp, un langage manipulant non plus des nombres, comme c'est le cas avec les langages conventionnels, mais des listes pour créer des procédures logiques.

Les langages utilisant des symboles (par exemple des mots ou des phrases), plutôt que des données numériques, et des opérateurs logiques plutôt que mathématiques sont appelés des langages symboliques.

Ce type de langage n'est bien sûr pas meilleur qu'un langage conventionnel, tout comme Cobol n'est pas meilleur que Fortran; tout dépend du problème lui-même, et/ou des hardware et software sur lesquels tourne le programme.

Le terme symbolique signifie que les faits, et la façon dont ils sont interreliés sont implémentés, non pas comme des nombres, mais plutôt comme des mots manipulés selon des règles logiques, c'est-à-dire en tirant des conclusions ou des inférences à partir de ces faits.

La figure 22 illustre les principales différences qui existent entre les langages symboliques et les langages conventionnels.

Précisons cependant quelques points :

- nous avons déjà distingué la principale différence : utilisation de symboles versus utilisation de données numériques ;
- les langages conventionnels sont basés sur des algorithmes, tandis que les langages symboliques reposent à la fois sur des algorithmes et sur l'heuristique. Par algorithme, il faut comprendre une suite d'instructions qui, exécutées, assurent l'obtention d'un résultat correct.

<u>Expert-Systems</u> <u>Terminology</u>	<u>Computer Science</u> <u>Terminology</u>
Knowledge engineer	Programmer-analyst
Knowledge-base	Program
Expert-system shell	Programming language
Knowledge-acquisition tool	Programming environment
Inference engine	Interpreter

Tableau 5 :
Les termes utilisés dans le domaine des systèmes experts
correspond étroitement à ceux utilisés en programmation
conventionnelle. [P.Hart, 1986]

Au contraire, l'heuristique dépend de l'expérience acquise au cours du temps, de l'existence de schémas mentaux, et donc n'entraîne pas nécessairement un résultat correct. Elle permet donc de réduire le nombre de noeuds à explorer dans l'arbre de résolution.

Bien qu'ils utilisent des algorithmes, quand ils sont disponibles, les langages symboliques ont recours à l'heuristique si la connaissance est incertaine, ou insuffisamment précise.

- les langages de programmation conventionnels sont basés sur un principe de répétition (exécution d'une même procédure une multitude de fois), alors que les langages symboliques dépendent de processus logiques ou inférentiels.

Bien que les langages conventionnels soient peu usités pour développer des systèmes experts, C et Pascal peuvent être employés - quoique secondairement - pour ce genre d'activité.

En ce qui concerne la terminologie utilisée, nous pouvons établir une correspondance entre les concepts utilisés dans ces deux approches de l'informatique. Le tableau 5 ci-contre présente d'autres points de comparaison possibles.

3.1.3. Programmation conventionnelle et Ingénierie de connaissances [RICH83], [HART86]

L'objectif de ce paragraphe est d'essayer de mettre en évidence les liens et les différences existant entre en ce qu'il est convenu d'appeler les méthodes de programmation traditionnelles et les systèmes d'intelligence artificielle dont font partie les systèmes experts. Notons cependant que certains points ont déjà été abordés implicitement lors de l'introduction aux systèmes experts (cfr. II.2.).

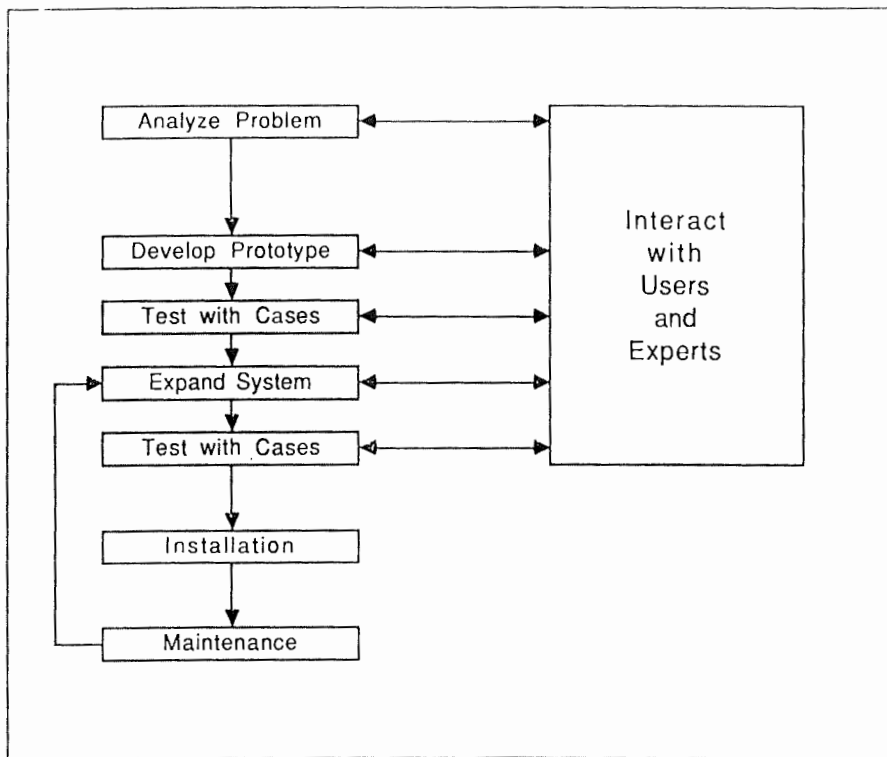


Figure 23 :
Phases de développement d'un système expert.
[P.Harmon et al., 1988]

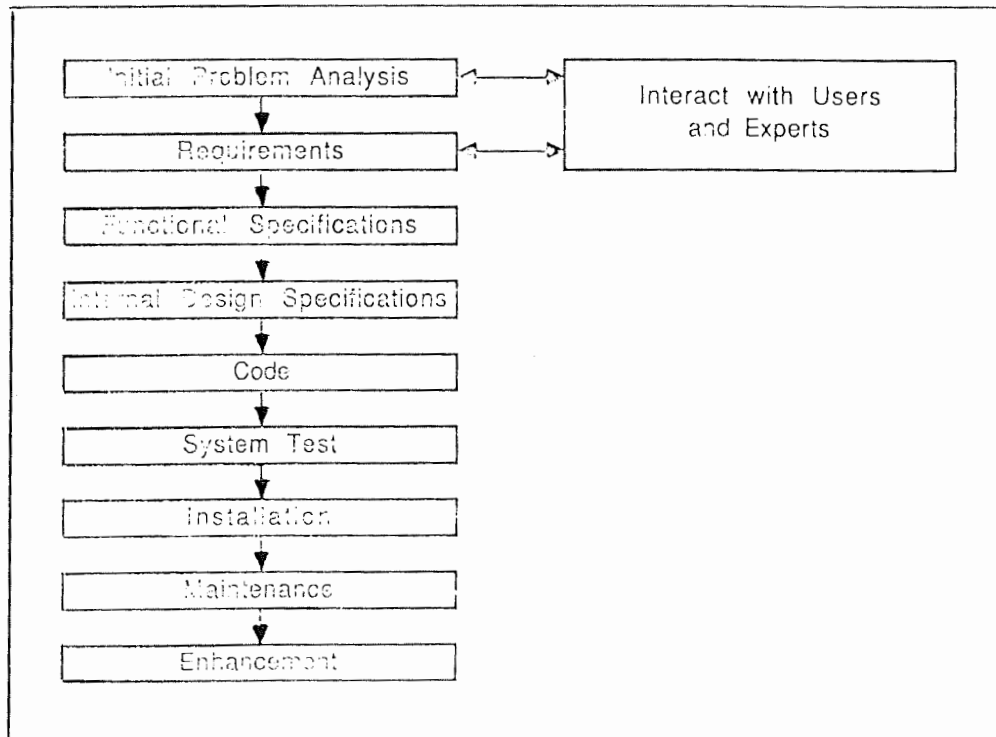


Figure 24 :
Développement d'un programme conventionnel versus ingénierie de connaissances.
[P.Harmon et al., 1988]

Les systèmes conventionnels sont développés séquentiellement (voir figure 23). Dans un premier temps, un analyste-programmeur rencontre un utilisateur et collabore avec ce dernier afin de développer une description fonctionnelle de la tâche susceptible d'être modélisée par le programme. A la fin de cette phase, l'analyste considère qu'il comprend suffisamment le travail pour réaliser le système.

Dans le processus d'implémentation qui en résulte, le programmeur construit un algorithme qui combine la connaissance acquise lors des rencontres avec l'utilisateur, et les instructions spécifiques au langage. Le code résultat est entièrement incompréhensible pour l'utilisateur. A ce moment, celui-ci a le choix entre accepter ce résultat ou demander des changements.

Les analystes-programmeurs qui sont accoutumés de travailler de cette manière, ont pour principe que le problème peut être défini une fois pour toute la durée de l'analyse. De plus, ils considèrent qu'une fois le problème compris, ils vont être seuls pour réaliser le code du programme. Il n'est pas question pour eux d'expliquer leur travail à l'utilisateur, étant entendu que ce dernier ne connaît rien à la programmation.

Le développement d'un système expert se base sur une approche totalement différente (voir figure 24). En effet, les ingénieurs de connaissances doivent s'attendre à travailler dans un domaine qui ne peut pas être bien délimité à l'avance. Le problème n'est pas que les experts humains n'expliquent pas ce qu'ils font, mais plutôt que la plupart du temps, ils ne savent pas l'expliquer. Ces mêmes ingénieurs doivent travailler patiemment en collaboration permanente avec les experts afin de développer et d'améliorer le système. Les cognitivistes doivent être également capables d'enseigner la technologie aux experts humains. De fait, un système expert efficace requiert fréquemment que l'expert, qui connaît la connaissance emmagasinée dans le système, peut suggérer des mises à jour et des corrections éventuelles. Cette caractéristique n'existe que dans la mesure où l'expert participe activement au développement.

En guise de conclusion, nous pouvons dire que dans l'approche traditionnelle, l'accent est mis sur l'analyse d'algorithmes de façon mathématique, ou, si cela est impossible, sur la réalisation d'une analyse statistique de leurs performances à partir d'un ensemble de problèmes soigneusement sélectionnés.

L'approche de l'intelligence artificielle pour l'étude d'algorithmes de résolution de problèmes est de coder les algorithmes, les faire tourner sur un ordinateur et d'observer leur comportement sur un échantillon de quelques problèmes. Deux raisons sont invoquées pour justifier une telle approche en intelligence artificielle : la première est qu'il est beaucoup plus "amusant" de voir un programme effectuer quelque chose d'intelligent, plutôt que de prouver qu'il peut le faire ; la seconde réside dans le fait que les problèmes soumis à une méthode de résolution en intelligence artificielle sont la plupart du temps complexes, au point qu'il est quasi impossible de produire une preuve analytique convaincante qu'un procédé tournera ou non. Toutefois, même si les résultats concernant la performance de tels programmes ne sont pas toujours significatifs, il ne faut pas pour autant les négliger.

Dans le point suivant, nous nous attarderons exclusivement sur les langages symboliques (Lisp, PROLOG), et certains langages orientés objet ; ils offrent les supports et la flexibilité souhaités pour une programmation en intelligence artificielle.

3.1.4. Lisp

Littéralement, ce nom provient de la concaténation de "LIST Processing".

Comme nous l'avons déjà exposé ci-dessus, Lisp travaille sur des symboles pouvant correspondre à toute combinaison de caractères; toutefois, il travaille essentiellement sur des suites telles que des mots (atomes), ou des phrases (listes).

Voici les idées clés de Lisp développées par son inventeur, J. Mc Carthy dans les années 1950-1960 : Lisp

- 1- travaille sur des expressions symboliques, plutôt que sur des nombres ;
- 2- représente les données comme des structures de listes interreliées au niveau interne, et comme des listes imbriquées (il y a plusieurs niveaux de listes) sur papier (niveau externe) ;
- 3- la structure de contrôle est basée sur une composition de fonctions afin d'en obtenir des plus complexes. On peut dès lors penser qu'une des raisons du choix de Lisp n'est pas l'utilisation d'un paradigme fonctionnel de programmation, mais bien l'utilisation de cet environnement (c'est-à-dire de l'ensemble des fonctions). Programmer en Lisp revient à étendre le langage en définissant de nouvelles fonctions ;
- 4- utilise la récursion pour décrire des processus et des problèmes ;
- 5- la fonction 'EVAL' : dès 1960, Mc Carthy décrivait 'EVAL' comme une fonction universelle qui sert d'interpréteur de toute fonction Lisp. Notons qu'à ce moment, Lisp n'existait pas encore ; c'était seulement une entité théorique. C'est sur base du codage de cette fonction qu'il a été possible de faire tourner le langage.

Lisp est un langage très interactif, flexible et récursif. Puisqu'il permet la récursion, Lisp aborde un problème en le scindant en plusieurs petits problèmes, ce qui simplifie grandement la résolution. Toutefois, la récursion doit avoir un

point de terminaison ! Ces propriétés ne facilitent peut-être pas la manipulation de ce langage (au point de vue syntaxique), mais permettent d'obtenir une solution à des problèmes complexes d'une façon plus rapide qu'au moyen de programmes écrits en langage conventionnel.

De plus, Lisp doit son succès au fait que le programme et les données ont le même format ; ainsi un programme peut utiliser d'autres programmes Lisp comme données. Une autre propriété liée à ce langage est la gestion automatique de la mémoire ; l'allocation de l'espace se fait de façon dynamique et facilite, allège donc le travail du programmeur. En fait, comme Lisp n'impose pas que les listes soient stockées de manière juxtaposées (vu l'existence de pointeurs sur les données), une telle gestion de la mémoire est efficiente.

D'autres particularités intéressantes, comme des fonctions de debugging, de compilation incrémentale, ainsi qu'une grande facilité d'extension sont offerts par l'environnement de la plupart des implémentations de Lisp actuellement disponible, et en font un langage très prisé en intelligence artificielle.

Malgré toutes les caractéristiques avantageuses évoquées jusqu'à présent, l'univers Lisp n'a pas toujours été rose. En effet, ce langage offre quelques fonctions de base à partir desquelles il est possible d'en définir d'autres, plus complexes. Celles-ci varient d'un dialecte Lisp à un autre (comme exemple de dialecte, nous pouvons citer Franz-Lisp, Mulisp, MacLisp, Interlisp,...). Ceci entraîne que le nombre de fonctions ainsi créées (pour l'ensemble des dialectes) est très important ; il est à l'origine d'une certaine redondance qui est à déplorer (plusieurs programmeurs implémentent chacun d'une manière différente une fonction ayant le même effet). C'est pourquoi, dès 1985, une nouvelle version, le Common Lisp, a été lancée sur le marché afin de standardiser ces fonctions.

Focalisons-nous à présent sur les caractéristiques de Prolog, autre langage symbolique.

3.1.5. Prolog

Prolog (PROgramming in LOGic), lui aussi bien classé au hit-parade des langages de l'intelligence artificielle, a été développé dans les années 1970 par A. Colmerauer et al. à l'université de Marseille. Ce langage implémente une version simplifiée du calcul des prédicats; programmer en Prolog revient donc à programmer en logique.

Ceci rend le langage idéal pour des applications qui requièrent une simulation de l'intelligence, y compris le développement de systèmes experts, le contrôle en robotique, les systèmes de planification et la modélisation.

Au lieu d'utiliser des concepts de programmation tels que "goto", "do-for",... Prolog a recours à des techniques comme le "pattern matching", la manipulation de listes, des bases de données relationnelles et la recherche en profondeur d'abord dans un arbre ET/OU basée sur un chaînage arrière.

Deux étapes sont essentielles lors du développement d'un programme Prolog :

- 1- spécifier des faits concernant des objets et les relations existant entre eux
- 2- poser des questions sur ces objets et ces relations (on formule des assertions et on demande si elles sont vraies) ; Prolog répond par "oui" ou par "non".

Notons que "oui" ne signifie pas que le prédicat est vrai, mais bien qu'il est prouvable en vertu des informations disponibles dans la base de connaissances (TIA, M. van Lamsweerde).

Détaillons brièvement ces deux propositions: pour représenter les objets relatifs au domaine particulier du problème à résoudre, ainsi que les relations qui existent entre eux, Prolog crée des clauses, ou règles de forme

`<tête> :- <queue>.`

où `<tête>` = la conclusion, le but
 `<queue>` = le(s) symptôme(s)
 `:-` = implication logique "`<=>`"
 (dans ce sens car on va toujours des symptômes vers le but et non l'inverse dans un problème de diagnostic) (TIA, M.van Lamsweerde).

Programmer en Prolog revient donc à poser des questions du style : "Quels faits et relations apparaissent dans ce problème ?" ou "Quelles clauses sont associées à ces faits et relations ?".

Puisque Prolog peut intégrer quelques caractéristiques extra-logiques comme des structures de contrôle (CUT) ou des opérateurs tels que `+`, `*`, `/`, `IS`,... pour des raisons d'efficacité (éviter l'explosion combinatoire du nombre de noeuds dans l'arbre de recherche), on ne peut classer ce langage comme purement et totalement logique.

Prolog présente deux sémantiques : déclarative et procédurale. La première concerne essentiellement la manipulation de procédures, et guide le système sur ce qu'il doit savoir, alors que la seconde considère le comportement de résolution que le programme doit adopter. Ceci est bien sûr transparent à l'utilisateur.

Naturellement, c'est la sémantique déclarative qui a contribué à rendre Prolog très populaire en intelligence artificielle, car cela permet de travailler sur des fragments de connaissances qui sont autonomes, explicites, et modulaires. De

ce fait, c'est un langage flexible, modifiable, vérifiable, et les fragments peuvent être réutilisables.

Pour des raisons de simplicité d'implémentation, l'ordre de considération des règles dans la base de connaissances est statique et prédéterminé en Prolog : de haut en bas pour les règles et les faits, et de gauche à droite pour les sous-objectifs dans une liste ET intervenant comme antécédents d'une règle.

C'est évidemment restrictif par rapport à d'autres systèmes qui recherchent de façon heuristique les règles et faits qui sont les plus prometteurs.

Prolog, comme nous l'avons déjà souligné, procède par recherche en profondeur d'abord en chaînage arrière dans un arbre représentant le contexte de travail. Le réel pouvoir de Prolog réside dans sa capacité à inférer (par déduction) des faits à partir d'autres.

Grâce à ces facilités, Prolog jouit d'une popularité internationale ; il est disponible sur des grands systèmes IBM et sur une large variété de PC's.

Il est intéressant de noter que, contrairement à Lisp, ce langage inclut un mécanisme d'inférence ; ceci peut constituer un avantage, mais peut également être défavorable dans le cas où l'utilisateur souhaite implémenter lui-même son moteur d'inférence (modules qui exécutent le raisonnement dans les systèmes experts). Sans doute, dans ce cas, préférera-t-il Lisp?

De plus, Prolog n'est pas vraiment standardisé.

Les deux langages symboliques que nous venons de décrire ne sont pas les seuls outils pour écrire des programmes en intelligence artificielle. Intéressons-nous donc à une autre catégorie de langages: les langages orientés objet.

3.1.6. Langages orientés objet

Un des premiers langages orientés objet fut Smalltalk, développé par Xerox Palo Alto Research Centers en 1976. Ecrit en Assembleur, Smalltalk a été utilisé pour construire des systèmes experts, mais son premier usage fut le développement d'environnements de programmation conviviaux, tels que des interfaces pour ordinateurs Xérox Star ou encore, plus tard, pour le Macintosh d'Apple.

Les concepts de base de la programmation orientée objet ont été incorporés à Lisp de différentes façons. Il existe donc, par exemple, des version Lisp d'un langage orienté objet ; il s'agit de Flavors, et encore Xlisp. Mais ces mêmes concepts peuvent également être implémentés en langages conventionnels ; ainsi, C++, environnement de programmation orienté objet, dérive de C.

Les deux concepts clés sous-jacents à la programmation orientée objet sont d'une part l'"encapsulation" de procédures et de données dans des entités appelées 'objets', et d'autre part l'héritage d'attributs et de comportements par la descendance d'un objet parent (existence donc d'une structure hiérarchique). En effet, chaque objet est vu comme un petit monde; il contient des programmes et des données, et peut interagir avec d'autres objets par échanges de messages lui indiquant ce qu'il doit réaliser, et comment le faire. La création d'un nouvel objet est souvent comparée au phénomène d'instanciation dans le sens où les nouveaux objets sont des instances de "vieux" objets.

En plus de la qualité des interfaces, les langages orientés objet offrent une grande flexibilité et une modularité importante lors du développement. De ce fait, de grandes firmes y ont de plus en plus recours : dans la nouvelle génération de PC's IBM, des environnements tels que Microsoft's Windows

rendent des techniques orientées objet disponibles pour les utilisateurs du DOS. [HARMON88]

Pour clôturer l'analyse de différents langages symboliques, ajoutons encore que les techniques actuellement disponibles pour développer des systèmes experts bénéficient d'un franc succès au point que les programmeurs manipulant des langages conventionnels se tournent de plus en plus vers les langages d'intelligence artificielle. Il va sans dire que la transition dépend de la disponibilité de nouvelles techniques qui intègrent les langages conventionnels et ceux de l'intelligence artificielle.

3.2. Outils de développement

3.2.1. Classification des outils

Dans cette section, nous allons donner un aperçu des programmes de manipulation de connaissances visant à résoudre un problème particulier - appelés outils de développement de systèmes experts ou shells - disponibles actuellement sur le marché.

La plupart des outils utilisés font une distinction entre la base de connaissances et le mécanisme d'inférence. De plus, on peut s'attendre à ce que le moteur d'inférence offre un certain nombre de fonctions ou interfaces qui rendent le développement de systèmes experts moins ardu et moins complexe qu'à l'aide de langages de programmation.

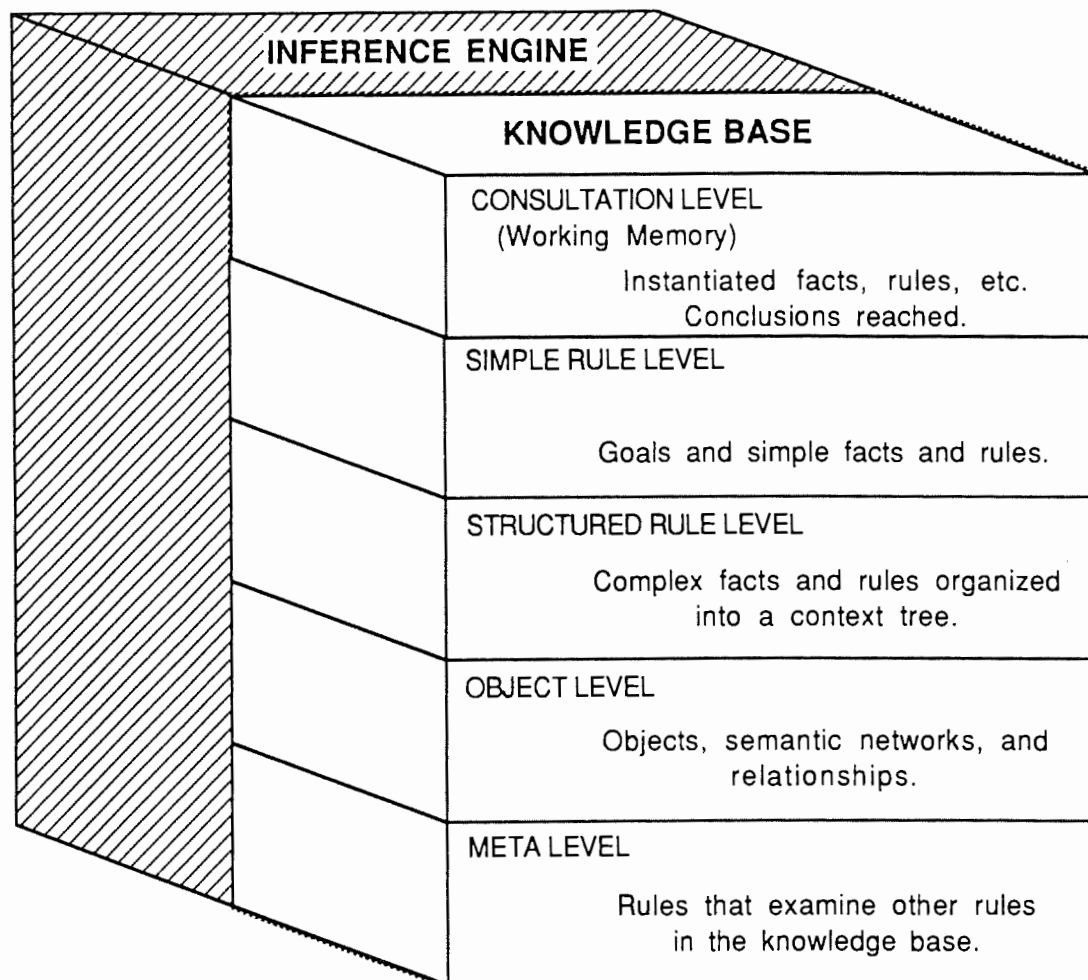


Figure 25 :
Moteur d'inférence et base de connaissances illustrant tous les
types de connaissances pouvant se retrouver dans une base de
connaissances durant une consultation.
[P.Harmon et al., 1988]

Une classification de shells peut bien sûr être réalisée de différentes façons. Celle que nous évoquerons ci-dessous est centrée sur les techniques de représentation des connaissances disponibles dans un outil, et s'établit de la manière suivante :

- outils d'induction (inductive tools)
- outils à base de règles simples ou structurées (simple or structured rule-based tools)
- outils spécifiques à un domaine ou dédiacés (domain specific tools)

La figure 25 illustre les différents types de connaissances pouvant exister dans une base de connaissances.

Nous allons maintenant passer en revue les différentes catégories d'outils.

a. Outils d'induction

Les outils d'induction génèrent des règles à partir d'exemples, ce qui limite fortement l'étendue des applications.

Le principe est le suivant : le développeur entre un grand nombre d'exemples dans la base d'informations, et par la suite, l'outil convertira à l'aide d'un algorithme les exemples en règles, et déterminera l'ordre que le système doit suivre en questionnant l'utilisateur.

Ces systèmes sont surtout efficaces dans des situations où de nombreuses données sont disponibles, et quand les connaissances relatives à un problème particulier peuvent facilement être représentées sous forme de règles capables de produire un résultat plausible. C'est le cas notamment dans le domaine du business.

Ces outils, dont font partie 1st-Class, TIMM, ou encore Super-Expert, sont surtout utiles pour représenter des connaissances relativement simples.

Il est possible de distinguer deux types d'outils : les "large inductive tools", tournant sur mainframes et PC's, et les "small inductive tools" qui eux ne tournent que sur PC's.

b. Outils à base de règles simples ou structurées

La classification de ce deuxième type d'outil peut se faire en fonction de la complexité des règles : elles sont dites simples ou structurées.

D'autres critères, comme la stratégie d'inférence (chainage avant ou arrière), les possibilités d'interfaçage (bases de données ou tableurs), le matériel cible (le nombre d'utilisateurs), et donc le prix (de 7000 FB à 2MFB) peuvent être pris en compte.

Les qualificatifs "simple" et "structuré" ne signifient pas que certains outils sont meilleurs que d'autres ; simplement, un outil est jugé sur sa capacité à résoudre un problème, et non sur les techniques qu'il offre. Ce qui veut dire que chaque type de problème requiert un type d'outil particulier en fonction de l'ampleur du "travail" à fournir pour le résoudre, ou du domaine d'expertise.

Dans le but de représenter les connaissances, les outils à base de règles simples utilisent des faits simples (de forme attribut-valeur), et des règles de type "IF...THEN...", manipulées en chainage arrière. Ils tournent sur PC's, et sont intéressants dans la mesure où les systèmes experts développés ne comptent pas plus d'environ 500 règles.

Dans cette catégorie, nous trouvons notamment VP-Expert, OPS5, Knowledge-Tool, Exsys, Insight-2,...

Comme nous allons le constater, les outils simples n'offrent pas autant de fonctions que les objets structurés. Toutefois, si les problèmes soumis sont mal structurés, l'efficacité de résolution par des outils plus sophistiqués et donc plus coûteux se réduit, et devient équivalente à celle des petits systèmes de règles.

Le genre d'applications souvent envisagé avec les petits shells tels que ceux que nous venons de décrire correspondent à des systèmes d'aide à l'accomplissement de tâches, car ils permettent, si l'on tient compte de leurs conseils, d'améliorer la qualité du travail réalisé, ainsi que d'accroître sa quantité. Ces systèmes d'aide consistent principalement en des "recettes" à suivre, ou des procédures "step-by-step", y compris des menus d'aide disponibles dans tout programme.

Comme nous l'avons déjà souligné, les outils à base de règles structurées offrent certaines caractéristiques qui ne se retrouvent pas dans les outils simples, et notamment des fonctions d'instanciations multiples, de représentation du contexte sous forme d'arbre, des facteurs de confiance, et des éditeurs plus puissants et plus complets. Les plus grands "rule-based tools" tournent sur mainframes, VAX, ou stations de travail UNIX. D'autres outils, plus petits, sont disponibles sur PC's. Ainsi nous trouvons Guru, Personal Consultant Plus, ESE,...

Leur fonctionnement est basé sur l'existence de faits complexes (contexte-attribut-valeur) et de règles de forme "IF...THEN..." organisées en ensembles vus comme des bases de connaissances séparées. Un ensemble de règles peut hériter d'informations lorsque d'autres règles d'un autre ensemble sont exécutées. Ces outils sont efficaces pour résoudre des problèmes importants par le nombre de règles requises pour la résolution, si ces règles peuvent être subdivisées en sous-ensembles distincts.

c. Outils hybrides

Ce type d'outils, dont font partie Goldworks, et KEE 386, représente l'environnement de développement de systèmes experts le plus complexe actuellement disponible sur le marché. De plus, c'est seulement depuis peu que de tels outils tournent sur PC's ; en effet, l'environnement sur lesquels ils tournaient auparavant étaient uniquement de type machines Lisp, VAX, et stations de travail UNIX configurées pour Lisp.

L'emploi de ces outils est basé sur la technique de programmation orientée objet. Un objet peut contenir des faits, des règles "IF...THEN...", ou des pointeurs vers d'autres objets.

La représentation des connaissances se fait donc au moyen de règles et d'objets structurés (classe, instance, attribut) d'où leur nom. On peut bien sûr énoncer d'autres caractéristiques telles que un mécanisme d'unification en principe plus complet, des stratégies d'inférence plus riches (combinaison des chainages avant et arrière), un raisonnement non monotone et au niveau des applications, ils facilitent grandement le développement d'interfaces utilisateur orientés graphique.

Puisque fort coûteux, l'emploi de tels outils ne se justifie que si le problème est réellement complexe ; difficiles à utiliser, ils requièrent une maîtrise du Lisp et de l'ordinateur sur lequel ils sont disponibles. De plus, il faut avoir la garantie de disposer d'une documentation suffisamment précise, gage d'un bon apprentissage.

En ce qui concerne les "large hybrid tools" comme KEE, ART et Knowledge Craft, ils ont été développés pour créer d'autres outils qui à leur tour construisent une base de connaissances ; ils sont donc plutôt considérés comme des outils "pratiques".

Ils représentent incontestablement les shells de l'avenir.

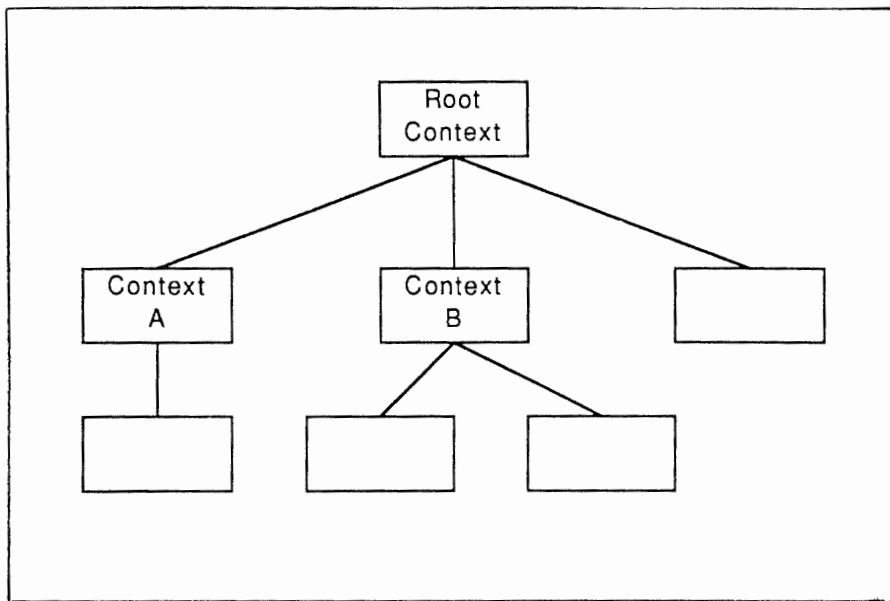


Figure 26 :
Modèle générique d'un arbre de contextes de type 'EMYCIN'.
[P.Harmon et al., 1988]

d. Outils dédiacés

Destinés à constuire des systèmes experts dans un domaine spécifique, les outils de cette cinquième catégorie sont conçus pour accueillir la connaissance de tout utilisateur potentiel dans un domaine précis.

En effet, grâce aux interfaces utilisateur et à un développement particuliers, ils permettent d'accélérer considérablement le développement de systèmes experts (ceci par rapport aux outils précités), alors qu'en réalité, ils peuvent inclure n'importe quelles caractéristiques définies dans ces autres outils (et donc "héritent" des mêmes propriétés).

C'est pourquoi, ils occupent une classe à part entière, et il semble qu'on puisse s'attendre à un accroissement relativement rapide de leur popularité dans les temps à venir.

Ainsi Maintex (outil destiné à la réparation d'équipements industriels (diagnostic, maintenance corrective)), et GPS (outil de gestion de procédures d'urgence) en sont quelques exemples.

3.2.2. Outils à base de règles versus outils orientés objet

Après avoir décrit brièvement les cinq catégories d'outils utilisables, il serait intéressant de préciser la distinction existant entre les outils "object-oriented" et les systèmes "rule-based".

Au départ, quand on créait une base de connaissances, on commençait par fixer un ou plusieurs "contexte(s)", et par y placer les règles.

Ces contextes sont reliés entre eux de façon à former un arbre, un contexte étant relié à un parent spécifique (voir figure 26).

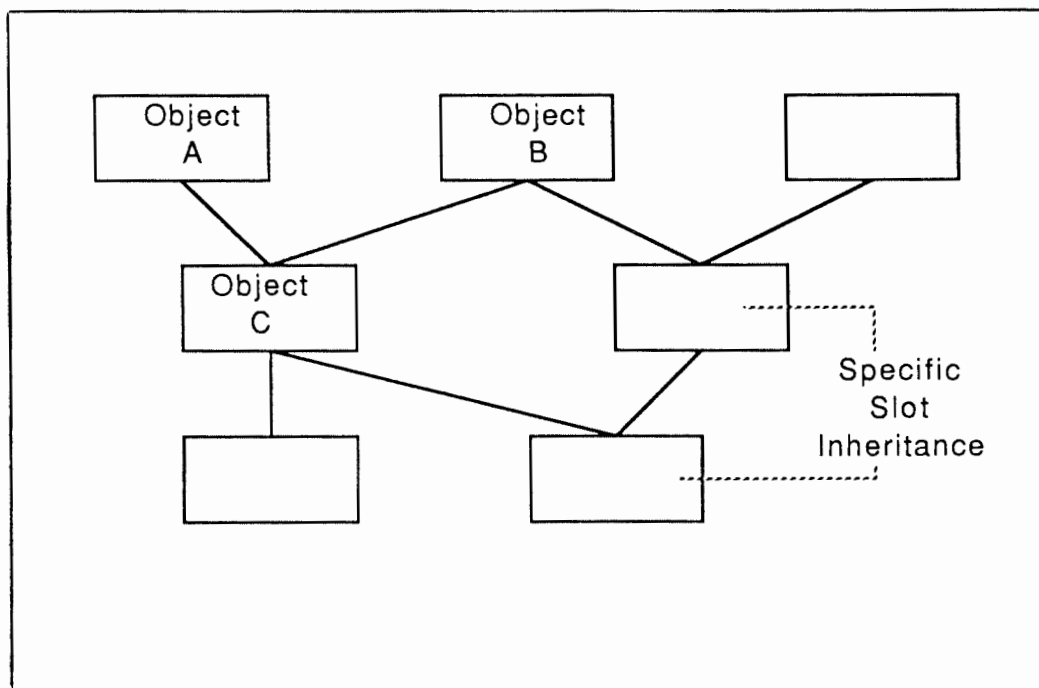


Figure 27 :
Un réseau objet.
[P.Harmon et al., 1988]

Une autre façon de voir les choses, apparue plus tard, est de parler non plus de contextes, mais de 'frames'. Le concept fondamental réside dans le fait que tout objet peut être mis en relation avec un autre par une variété de chemins différents. Un système orienté objet correspond donc à un réseau complexe d'objets pouvant hériter d'informations provenant d'autres objets, et ce d'une manière plus complexe, plus diversifiée que dans les systèmes de règles (voir figure 27).

Dans une autre perspective, quand on construit un système 'rule-based', on se focalise sur les règles, les procédures en quelque sorte, nécessaires pour résoudre le problème. Par contre, à l'aide d'un outil orienté objet comme KEE, l'accent est mis sur la description d'objets extraits du domaine d'expertise, ainsi que sur les relations qui existent entre ces différents objets et leurs composants.

Après avoir abordé d'une façon globale les langages et outils utilisés en intelligence artificielle, nous avons jugé intéressant d'étudier plus en détails trois shells, à savoir VP-Expert, Nexpert et Guru.

3.3. Analyse de trois outils : VP-Expert, Nexpert et Guru

Une question peut évidemment venir à l'esprit du lecteur : pourquoi avoir opté pour ces trois shells? Il est vrai que ce ne sont pas spécialement les outils les plus prisés par les différentes sociétés, encore que leur usage soit très fréquent. Précisons qu'une des raisons de notre choix vient du fait qu'ils sont disponibles à l'Institut d'Informatique, et que dès lors, il était intéressant de procéder à une analyse comparative.

Nous relèverons tout d'abord différents critères que nous avons pris en compte pour ensuite distinguer les trois outils.

Pour rédiger cette partie, nous nous sommes basés sur les chapitres 5 à 9 de l'ouvrage de P.Harmon, R.Maus, W.Morrissey : Expert Systems Tools and Applications, ainsi que les manuels d'utilisation des trois shells.

3.3.1. Critères d'analyse

Dans le but d'évaluer un outil de développement de systèmes experts, différentes caractéristiques peuvent être prises en compte.

a. Représentation des connaissances, inférence et contrôle

Ce critère décrit :

- le type de connaissances (problèmes) que l'outil peut manipuler
ex : règles et faits simples,
faits complexes, règles, et arbres pour visualiser le contexte, objets, réseaux sémantiques (voir figure 25) ;
- la stratégie d'inférence du moteur (chainage avant ou arrière, manière selon laquelle il convient donc de représenter les informations dans la base de connaissances) (cfr point 2.4.1. chap. 3) ;
- les facteurs de confiance ou coefficients de certitude : ce sont des valeurs associées aux faits et règles indiquant le degré de confiance que l'expert a dans une règle, ou que l'utilisateur a dans un fait. Ce sont des valeurs empiriques : elle correspond à une quantification d'une appréciation en fonction de l'application ;

- l'héritage : c'est la stratégie d'inférence des systèmes orientés objets (créer de nouveaux faits à partir d'autres existant déjà ; un objet créé hérite des faits, des comportements du (des) parent(s)).

b. Interface développeur

Ce critère fait référence aux capacités de l'outil concernant la construction et les modifications de la base de connaissances, étapes essentielles pour une bonne implémentation.

On peut donc y distinguer :

- la création de la base de connaissances via un éditeur interne ou externe ;
- l'explication du raisonnement suivi : messages à inclure par le développeur pour qu'ils soient disponibles pour l'utilisateur ;
- "How" (l'utilisateur ou le développeur peut se demander comment le système a atteint une conclusion) et "Why" (il peut chercher à savoir pourquoi une question lui est posée à un moment précis de la consultation) : ces options sont-elles présentes ?
- la trace du raisonnement suivi : fait référence aux inférences faites par le système durant une consultation. Cette trace notamment utile pour la construction de la réponse au "How" ;
- les facilités offertes par l'éditeur.

c. Interface utilisateur

- les menus ;
- les liens possibles avec d'autres packages pour construire des graphes (en effet, il existe très peu d'outils qui permettent de développer des graphes dans le shell lui-même sans faire appel à d'autres logiciels).

d. Interface système

Ce critère concerne la manipulation à partir de l'outil d'informations ou langages contenus dans des fichiers externes à la base de connaissances.

On y distingue :

- "Fast", "Slow" : permet d'augmenter ou diminuer la vitesse d'exécution à la consultation ;
- l'accès à des bases de données et tableurs externes ;
- l'appel à des routines externes écrites dans un langage différent.

e. Software

- langage avec lequel est écrit l'outil ;
- OS requis (DOS sur PC's, et UNIX, Lisp, VMS,... sur mini-ordinateurs ou mainframes)

f. Hardware

PC-XT, PC-AT, minis, mainframes, stations de travail, machines LISP,...

g. Documentation

relative à la manipulation de l'outil.

Dans l'analyse qui suit, la plupart des critères énoncés ont été pris en compte.

3.3.2. Analyse comparative

a. VP-Expert

VP-Expert est un outil de haute qualité qui combine des propriétés d'outils plus coûteux et plus performants, et des nouvelles caractéristiques qui lui sont propres. Lancé sur le marché en 1986, il est considéré jusqu'à présent comme un des meilleurs outils disponibles sur PC's ; de plus, son prix est relativement faible.

VP-Expert représente les faits comme des doublets (attribut-valeur), et ces attributs peuvent correspondre à des variables indicées (ex : var[1], var[2],...). Cette caractéristique est utile, surtout si on désire travailler avec le contenu de plusieurs cellules ou enregistrements issus respectivement d'un tableur ou d'une base de données.

Le chaînage arrière est pour ainsi dire la seule stratégie de recherche sur laquelle repose le fonctionnement du moteur d'inférence. Toutefois, certaines instructions placées dans la conclusion d'une règle peuvent le forcer à travailler en chaînage avant, quoique celui-ci fût limité.

Remarque : ceci est tiré de l'ouvrage Expert Systems Tools and Applications, mais n'est pas confirmé dans le manuel d'utilisation dont nous disposons.

La plupart des petits outils à base de règles n'utilisent pas les facteurs de confiance (CF ou CNF), ou les implémentent en se basant sur des probabilités standards, ce qui les rend peu crédibles.

Pour ce qui est de VP-Expert, il offre une implémentation de ces CF comme le font d'autres outils plus sophistiqués. Ainsi, il utilise des expressions mathématiques au niveau des prémisses ou des conclusions de règles.

Les valeurs de ces CF peuvent être introduites dans les conclusions de règles au niveau de la base de connaissances, ou entrées par l'utilisateur en réponse à une question accompagnée d'un menu. Elles peuvent s'échelonner de 0 (%) à 100 (%) ; toute valeur inférieure à 50 entraîne une réponse négative. La valeur par défaut est de 100.

Notons que la clause 'TRUTHTHRESH' permet de modifier le seuil de vérité de 50 à toute autre valeur comprise entre 0 et 100.

Comment en réalité sont calculés les CF? Partons d'un exemple :

```
IF    X = A
THEN T = D  CNF 80
```

La valeur '80' du CNF de la conclusion suppose que $CNF(A) = 100$.

Si on a $CNF(A) = 50$, dans ce cas $CNF(D) = 0.50 * 0.80$
 $= 0.40$ (ou 40 si on travaille en pourcentages)

Et si seul CNF(A) a été précisé et vaut 75 par exemple,
alors $CNF(D) = 75$ également car $0.75 * 1.00 = 0.75$.

Toutefois, ce n'est pas toujours aussi trivial. Distinguons trois cas :

1. la condition est composée d'éléments connectés par
AND : dans ce cas, le CNF associé à la variable but
de la conclusion est calculé en multipliant le plus
petit CNF qui apparaît dans la condition par le CNF
original de la variable but.

```
ex :  IF    X = A and
        Y = B and
        Z = C
      THEN T = D   CNF 70
```

```
avec CNF(A) = 50
      CNF(B) = 70
      CNF(C) = 60
nous obtenons CNF(D) = 0.50 * 0.70
                    = 0.35   ou 35 %
```

2. la condition est composée d'éléments connectés par OR:
la formule de base est :

$$CNF1 + CNF2 - (CNF1 * CNF2) = CNF \text{ (variable but)}$$

où CNF1 et CNF2 = CNF calculés pour les deux
expressions logiques apparaissant dans la
condition de la règle

```
ex :  IF    X = A or
        Y = B
      THEN T = D   CNF 80
```

```
avec CNF(A) = 80
      CNF(B) = 50
```

nous obtenons $CNF1 = 0.80 * 0.80 = 0.64$
 $CNF2 = 0.50 * 0.80 = 0.40$
 et donc $CNF(D) = 0.64 + 0.40 - (0.64 * 0.40)$
 $= 0.78$ (ou 78 %)

3. la condition est composée d'éléments connectés par AND et OR : le CNF de la variable but est calculé comme s'il y avait plusieurs règles.

ex : IF X = A and
 Y = B or
 Z = C
 THEN T = D CNF 80

ceci correspond à IF X = A and
 Y = B
 THEN T = D CNF 80

plus IF X = A and
 Z = C
 THEN T = D CNF 80

avec $CNF(A) = 60$
 $CNF(B) = 70$
 $CNF(C) = 50$

nous obtenons $CNF1 = 0.60 * 0.80 = 0.48$
 $CNF2 = 0.50 * 0.80 = 0.40$
 et donc $CNF(D) = 0.40 + 0.48 - (0.40 * 0.48)$
 $= 0.69$ (ou 69 %).

Dans le but de créer la base de connaissances, on peut avoir recours à l'éditeur interne de VP, ou à un éditeur de texte comme WordStar. De plus, VP-Expert offre la possibilité d'obtenir des règles à partir d'exemples stockés dans une table d'induction, dans un fichier BD ou un tableur (via un "inductive frontend").

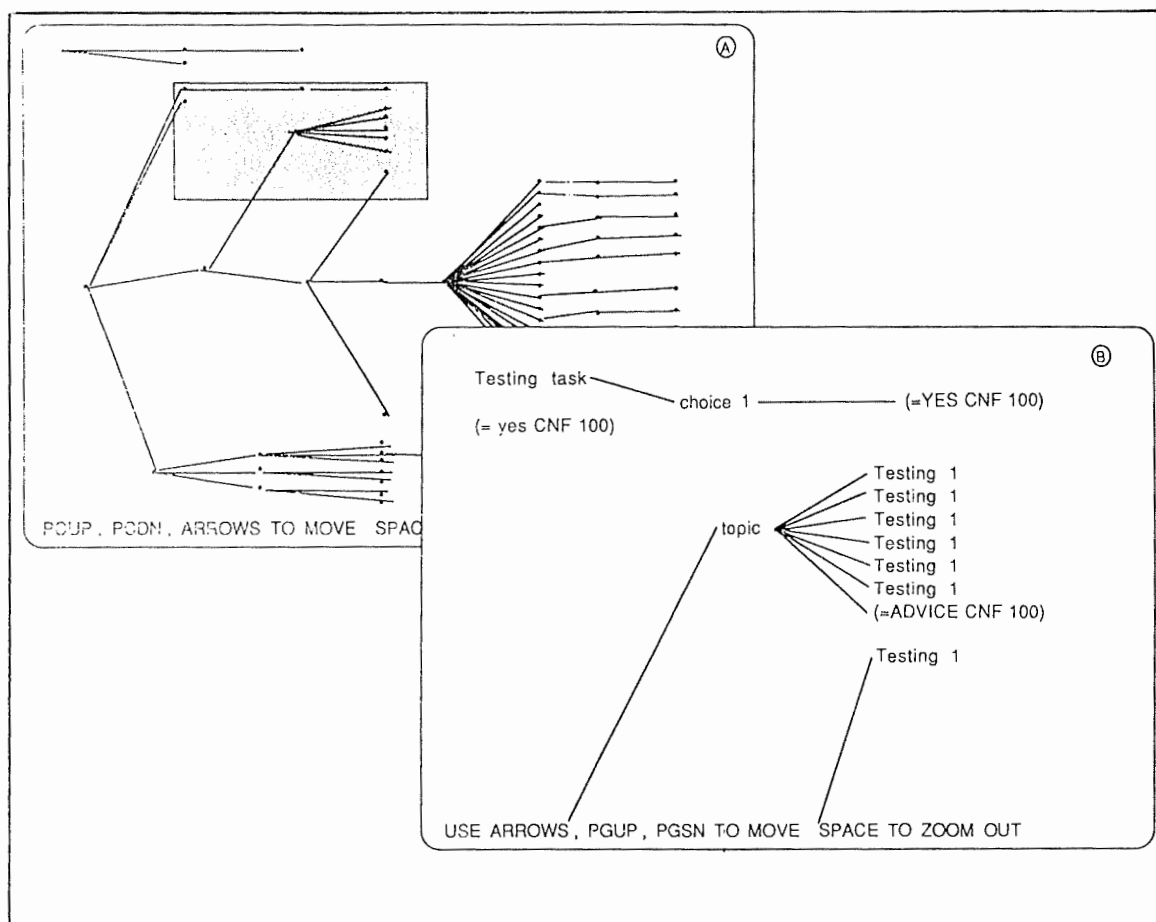


Figure 28 :
Ces deux écrans illustrent la trace graphique d'une consultation :

(A) montre comment apparaît l'écran

(B) est un "focus" sur la surface hachurée.

[P.Harmon et al., 1988]

Les règles sont représentées sous la forme "IF...THEN..." ou "IF...THEN...ELSE...". Les différentes composantes de la condition peuvent être reliées par les opérateurs logiques AND et/ou OR ; dans les conclusions, il peut y avoir toute une série d'actions à effectuer, exactement comme dans le bloc ACTIONS (cette notion sera expliquée ci-dessous). Il est possible d'insérer des règles où l'on veut ; l'ordre n'a de répercussions que sur la vitesse d'exécution du programme. La flexibilité est donc importante.

Un élément central dans la base de connaissances de VP est le bloc ACTIONS qui identifie les actions à exécuter durant une consultation. Les clauses, composantes du bloc, sont exécutées dans l'ordre où elles apparaissent.

Une fois qu'une consultation est amorcée, l'utilisateur peut, s'il le désire demander au système des explications concernant le raisonnement suivi. En effet, une clause 'BECAUSE', insérée par le développeur en fin de règle, permet de présenter un message explicatif en réponse à la commande 'WHY' ou 'HOW' lancée par l'utilisateur.

Dans le cas où 'BECAUSE' n'apparaît pas dans la règle, c'est la règle elle-même qui est affichée.

Il est également possible, une fois la consultation terminée, de visualiser la "trace" des inférences faites par le système sous forme d'un arbre ou d'un texte (voir figure 28). Une option, 'WHATIF', permet de découvrir comment la dernière consultation serait affectée si l'utilisateur avait répondu autrement à la question posée.

VP-Expert possède une des interfaces utilisateur les plus conviviales qu'un outil offre sur PC's ; un menu est présenté au bas de l'écran, dévoilant ainsi différentes options disponibles. Des sous-menus sont alors accessibles à partir de là. Notons l'absence d'une souris qui faciliterait l'interaction avec l'utilisateur.

La sélection est aisée vu l'utilisation de contrastes (brillance) au niveau des menus. De même, lors d'une

Do you want to know how to control the order in which events occur during the consultation?

NO ◀ YES

Are you seeking help on how to call external programs, how to share data with external programs or neither?

NEITHER ◀ CALLS SHARING DATA

Do you want to know how to store information in a database, or how to retrieve it?

STORING RETRIEVING ◀

1 **Help** 2Go 3Whatif 4Variable 5Rule 6Set 7Quit
Help the user

Figure 29 :
Cet écran illustre la consultation que VP-Expert lance suite à une demande d'aide de l'utilisateur.
[P.Harmon et al., 1988]

consultation, il suffit à l'utilisateur de "jouer" avec les flèches afin de se positionner sur la réponse de son choix. Un exemple est donné par la figure 29 illustrant la sélection du programme d'aide.

De plus, en l'absence de l'assertion 'RUNTIME' dans la base de connaissances, trois fenêtres apparaissent à l'écran durant une consultation. Il est effectivement intéressant de pouvoir observer ce qu'il se produit dans la "tête", dans l'intelligence du système surtout pendant la phase de développement d'un programme.

Ainsi, la fenêtre 'RULES' fournit une vue locale, ponctuelle de la base de connaissances (on peut de ce fait se rendre compte de la façon dont le moteur interagit avec la base de connaissances) ; la fenêtre 'RESULTS' affiche quant à elle les conclusions intermédiaires et/ou finales sous la forme d'équations (ex : variable = valeur CNF x), alors que la troisième fenêtre correspond à l'écran de consultation présenté en modèle réduit !

A noter que le résultat de la commande 'FAST' ou 'SLOW' (exécution plus rapide ou plus lente du programme) peut être visualisé sur les fenêtres 'RULES' et 'RESULTS' (défilement des règles plus ou moins rapidement).

A partir de ces menus, il est également possible d'examiner une ou plusieurs règles choisies (indépendamment de la fenêtre 'RULES'), ainsi que les valeurs prises par une variable déterminée durant une consultation.

Dans VP-Expert l'interface système est également remarquable. Nous en avons d'ailleurs précisé un aspect ci-dessus en mentionnant les options 'FAST' et 'SLOW'. Mais ce qui est plus important, c'est de pouvoir interagir avec une base de données ou un tableur au cours d'une consultation.

VP peut notamment interagir avec dBASE II, III et III+. La clause 'GET' permet de retirer des éléments de la base de données et de les assigner à des variables utilisables dans le programme. Les clauses 'PUT' et 'APPEND' permettent quant à

elles de transférer des données dans le sens inverse, c'est-à-dire de VP-Expert vers une base de données.

'PUT' est utilisé pour mettre à jour des enregistrements existant, tandis que 'APPEND' permet d'en ajouter de nouveaux. Etant donné que VP offre la possibilité à l'utilisateur de modifier des records et d'en ajouter et ce, durant une consultation, on peut dire en quelque sorte que le système expert est capable d'apprendre par "expérience".

De plus, VP-Expert peut également échanger des informations avec différents tableurs tels que VP-PLANNER, Lotus 1-2-3,... Les modalités d'interaction se font essentiellement grâce à deux clauses : 'WKS' et 'PWKS'.

La première permet un transfert de données des cellules du tableur vers la base de connaissances ; la seconde effectue le transfert en sens inverse.

Toutefois, lors de notre travail, il est apparu que des données mises à jour dans Lotus 1-2-3 ne peuvent pas être manipulées par le programme lors de la même consultation ; il semble qu'une réinitialisation du programme soit nécessaire avant de pouvoir utiliser ces données.

A noter qu'en une seule commande, le développeur peut recueillir le contenu de plusieurs cellules ou de plusieurs enregistrements. Beaucoup d'outils, tel que Guru, ne permettent d'accéder qu'à une seule cellule ou record à la fois suite à une instruction.

En plus de sa capacité d'interagir avec des bases de données et des tableurs, VP-Expert peut aussi appeler des fichiers externes. Cette caractéristique lui permet d'exécuter les fonctions et opérations d'autres logiciels.

Quatre clauses rendent cette interaction possible :

- CALL est utilisé pour exécuter des fichiers DOS avec comme extension .EXE,
- CCALL est utilisé pour appeler des fichiers DOS dont l'extension est .COM,
- BCALL exécute des fichiers DOS .BAT,
- WORKON permet d'appeler un tableur donné à l'écran.

En ce qui concerne les caractéristiques logicielles inhérentes à VP, nous pouvons dire qu'il est écrit en C, et que l'OS requis est le DOS, version 2xx ou 3xx. Du point de vue hardware, il ne tourne que sur les IBM PC's et compatibles.

Guru et Nexpert, les deux autres outils que nous allons développer, se classent parmi les outils de taille moyenne, et ce d'une part par rapport à des outils hybrides et quelques plus grands outils (dits "large") à base de règles structurées, et d'autre part par rapport aux outils d'induction et aux outils à base de règles simples (dits "small"). Nous allons tout d'abord étudier Nexpert.

b. Nexpert

Nexpert (version 0.96) est un outil lancé par Neuron Data en 1985 ; il est très bien adapté à l'environnement graphique et hautement interactif du Macintosh. A première vue, il présente de nombreuses ressemblances avec la version de Nexpert tournant sur des stations de travail Lisp. A l'origine développé en Lisp, Nexpert a été converti en un mélange de C, de Pascal et d'assembleur, et tourne actuellement sur Vax.

Depuis 1987 deux nouvelles versions de Nexpert sont apparues (d'abord 1.0 et ensuite 1.1 en 1988). Nexpert-Object, le nom du dernier né, est un outil hybride puisqu'il se sert de règles (pour représenter le raisonnement) et d'objets (pour décrire l'univers sur lequel le raisonnement est fondé) pour représenter les connaissances. Il existe en réalité un certain degré d'interdépendance entre ces deux dimensions orthogonales (raisonnement - qui correspond à la façon dont l'information est manipulée - et représentation). Cette dépendance repose sur le fait que, d'une part les règles manipulent des objets (création, suppression,

modification), et d'autre part les objets peuvent "contenir" des règles dans le sens où ils peuvent en forcer le déclenchement. Ainsi le méta-slot 'If Change' définit les actions à effectuer si la valeur d'une propriété d'un objet ou d'une classe est modifiée (exemple d'actions : affecter le processus d'inférence, afficher la modification à l'écran, ou encore envoyer un message à un autre objet et/ou classe).

Cette méthode, comme tous les méta-slots, peut-être héritée.

Un objet est une unité de description (toute "chose" du réel est un objet) qui est définie par ses nom, classe(s), sous-objet(s), propriétés, et méta-slots.

Ils sont répartis en classes ; ces deux entités ont la possibilité de transférer dynamiquement - c'est-à-dire durant une consultation - des fonctions (des comportements) et des valeurs d'un objet/classe vers un(e) (des) autre(s) objet(s)/classe(s). Nous pouvons donc observer que l'héritage est une notion relativement vaste dans la mesure où il est possible de distinguer l'héritage descendant et ascendant ; de plus, des niveaux de priorité ('inheritance category') peuvent être assignés à toute propriété de n'importe quel objet ou classe. Une comparaison de ces valeurs permet de résoudre des conflits dans le cas d'héritage multiple (un objet peut effectivement hériter de valeurs de plusieurs parents). La propriété d'héritage peut être définie par l'utilisateur via les méta-slots.

Pour Nexpert-Object, les règles - modélisant les relations d'inférence - représentent donc la dimension du raisonnement ; cependant, une autre notion doit être évoquée au niveau de l'inférence. Certaines connaissances, l'intuition par exemple, ne peuvent pas être codées sous forme de règles ; le système utilise alors des contextes. Ce mécanisme est à la base des "îles de connaissances", ensembles de une ou plusieurs règle(s) reliées entre elles par des liens faibles. Ceci entraîne qu'une règle est explorée soit indépendamment de toute autre règle (c'est-à-dire directement), soit automatiquement si une "voisine" l'a été.

Nexpert, déjà dans sa version 0.96, introduit une nouvelle et puissante stratégie d'inférence qui permet au concepteur d'écrire les règles sans se préoccuper d'un chaînage avant ou arrière. Pendant une consultation, le système peut user de l'une, de l'autre, ou même des deux stratégies "à la fois". C'est à l'utilisateur qu'incombe le choix de l'une ou l'autre stratégie. Par défaut, c'est le chaînage mixte qui est d'application. Celui-ci permet d'utiliser un ensemble de connaissances de façon optimale.

Remarquons que, pour qu'un chaînage mixte soit applicable, il est nécessaire que toutes les règles soient écrites de façon à pouvoir être utilisées de cette manière ; c'est pourquoi dans Nexpert, les règles sont dites "symétriques".

Contrairement à M1 (un outil de la même catégorie), le chaînage avant offert par Nexpert (0.96) peut utiliser une règle plus d'une fois, mais il n'est pas capable d'instancier une même règle à plusieurs objets (pas d'instanciations multiples). Ceci entraîne que si un groupe de règles doit être exécuté n fois lors d'une consultation, il doit être écrit n fois. Nexpert-Object, quant à lui, présente des méta-règles contrôlant explicitement d'autres règles ; elles permettent notamment de contrôler l'exécution d'une boucle, ... Dans ce cas, une même règle peut être exécutée plusieurs fois ; il faut toutefois veiller à réinitialiser à une valeur inconnue les variables qui apparaissent dans la conclusion (c'est la fonction du 'RESET').

Nexpert-Object autorise maintenant que son système n'utilise qu'une seule règle pour "scanner" une liste d'objets groupés dans une conclusion (cette liste est construite lors de l'évaluation de l'ensemble des conditions). Cette caractéristique est donc à l'origine d'une réduction du nombre de règles, et ce par rapport à la version 0.96. Ceci débouche sur la notion de "pattern-matching" où plusieurs objets peuvent être traités en une seule inférence.

Posons-nous la question de savoir comment sont représentées les connaissances dans Nexpert. Les faits sont représentés comme des paires (attribut-valeur). Exemple : "le gel est le SEPHADEX G50" est représenté par "gel = SEPHADEX G50". Ces faits sont introduits dans des règles de forme

"IF... THEN...and DO..."

où les éléments de la condition sont connectés via les opérateurs booléens AND ou NOT, mais pas OR alors qu'il est accepté par d'autres outils de la même catégorie, comme M1. Cet opérateur est réalisé en écrivant deux règles différentes (ou plus) ayant pour condition la (les) condition(s) reliée(s) par OR. 'THEN' est suivi d'une hypothèse (ou but, conclusion) qui devient vraie si la (les) condition(s) est (sont) vérifiée(s), 'DO' contient un ensemble d'actions à effectuer si la règle est évaluée à "vrai".

Au niveau de la base de connaissances, l'accent est mis surtout sur la notion de "focalisation de l'attention" et d'allocation de "ressources", plutôt que sur une approche algorithmique.

Un problème est donc considéré, non pas comme un état, mais comme un événement qui requiert l'allocation de "ressources". Pour Nexpert-Object, la dimension humaine (l'implication de l'expert) semble donc centrale dans la définition du problème. Plutôt que d'utiliser une logique formelle comme base à la structure des connaissances, Nexpert-Object exprime la connaissance par des 'frames' prenant en compte l'intuition et les connaissances incomplètes (voir notion de contexte). Une règle est donc une unité de cognition qui détermine une direction à suivre lors de la résolution du problème.

Dans la version 0.96, les règles qui ont été écrites restent groupées en un seul ensemble.

Par contre, Nexpert-Object offre la possibilité d'avoir plusieurs modules distincts ou bases de connaissances.

En effet, une fois que les différents fragments de connaissance (règles, classes, objets,...) sont édités, ils sont sauvés dans un ou plusieurs fichiers.

Les faits sont caractéristiques d'une application particulière, et donc peuvent constituer l'input d'une session (exemple : pour établir un diagnostic, il est nécessaire d'introduire des symptômes). Pour des applications plus complexes, la plupart des faits peuvent provenir de fichiers sur disque ou de bases de données (au sens large).

En ce qui concerne les facteurs de confiance, Nexpert n'est pas aussi nuancé que le sont d'autres outils. En effet, toute connaissance est soit vraie, soit fausse, alors que M1 et Personal Consultant Plus chiffrent l'incertitude de -100 (information fausse) à +100 (information vraie). Cependant, avec Nexpert, il est possible d'insérer dans le programme des règles dont la fonction est de calculer un degré de confiance ; ceci ne paraît pourtant guère réalisable avec des systèmes importants, car il faudrait que la formule de calcul se retrouve dans chaque règle manipulant des informations incertaines.

De plus, il est également possible d'associer des priorités (via le 'category editor') à certaines règles, et donc indirectement à des valeurs de variables ; ceci équivaut en quelque sorte à un degré de certitude artificiel.

En effet, au moyen de ce 'rule category', le mécanisme d'inférence est guidé vers la règle de priorité maximale, avant de considérer les suivantes (par ordre de priorité décroissante). L'éditeur permet donc de contrôler l'ordre dans lequel les événements se produisent. Dans ce cas, on dira que plus une règle est prioritaire, plus elle est certaine (du point de vue probabiliste).

Ces priorités peuvent également être associées à des données, expressions, hypothèses, conditions,...

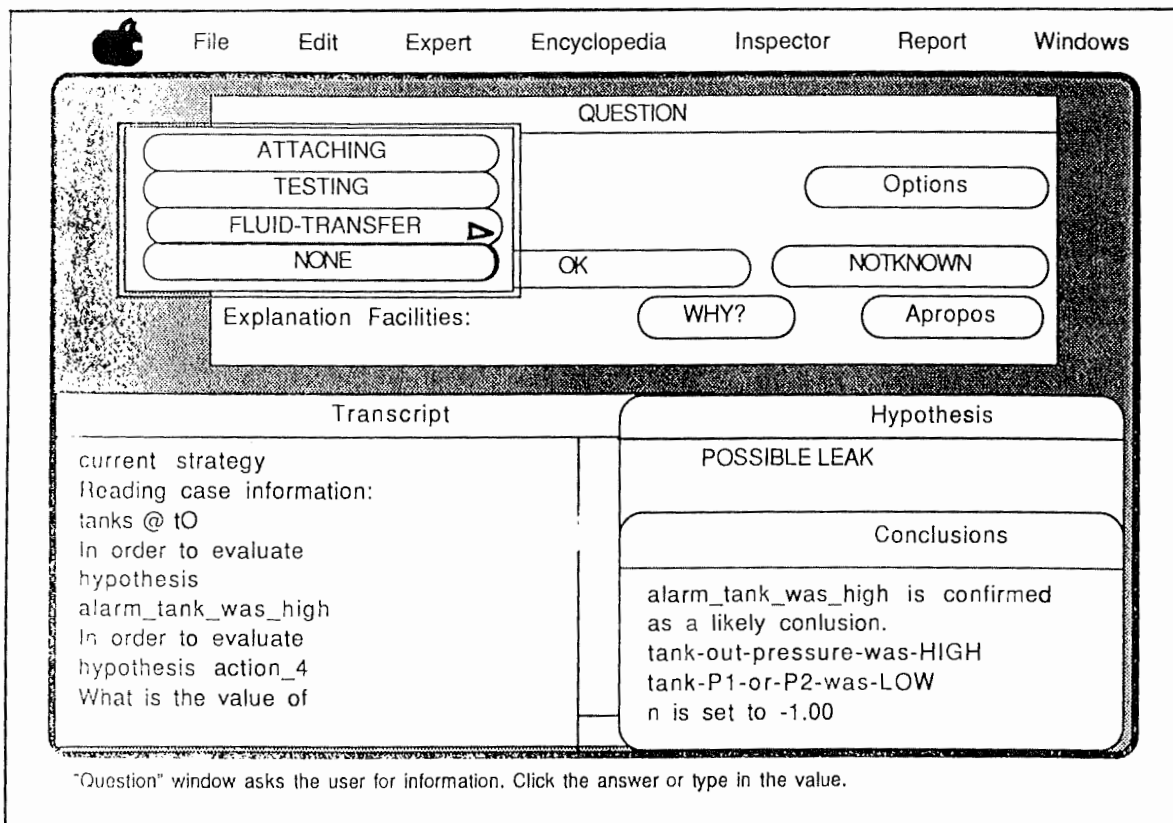


Figure 30 :
Nexpert : écran de consultation pendant la phase de développement.

[P.Harmon et al., 1988]

La phase de développement est aisée dans la mesure où l'édition de règles et d'autres connaissances se fait sans quitter le programme. De même, l'entrée de réponses aux questions se fait tout aussi facilement à partir du clavier ou d'une souris.

Le réseau de règles ainsi formé peut être visualisé sous forme de graphe, ce qui donne une perspective de plus haut niveau à l'information.

Durant une consultation, le développeur a l'occasion de suivre le cheminement du système sur deux fenêtres, l'une indiquant les règles choisies, et l'autre montrant les conclusions tirées.

De plus, la possibilité d'intégrer des graphes lors d'une consultation est intéressante, surtout quand l'utilisateur final n'est pas familier au domaine d'expertise ; en effet, cela permettrait d'illustrer le contexte d'une question posée ou une conclusion que le système a établi.

Nexpert met également à la disposition du programmeur un moyen d'exécuter des routines externes via un appel à l'interface externe ('Callable Interface').

Du point de vue utilisateur, Nexpert est très interactif. Une consultation s'effectue en trois étapes :

1. le système demande un signal d'initialisation à l'utilisateur. Comme Nexpert travaille en chaînage avant et/ou arrière, l'utilisateur doit entrer soit des faits (pour lancer le chaînage avant), soit des hypothèses (dans le cas de chaînage arrière). Ceci est assez compliqué dans le cas où l'utilisateur est étranger au domaine, mais peut être efficace dans le cas contraire.
2. l'utilisateur répond aux différentes questions posées par le système.
3. le système affiche les conclusions. A l'exécution, il est possible à l'utilisateur de demander comment une conclusion a été obtenue ('HOW'), et pourquoi une question lui a été posée ('WHY') (voir figure 30).

Notons que l'utilisateur ne peut pas associer de facteur de confiance aux réponses qu'il introduit ; ceci est cependant permis avec d'autres outils tels que M1.

A noter que dans le cas d'une application développée avec Nexpert, aucune interface utilisateur n'est à proprement parlé disponible. Elle doit être créée séparément à l'aide d'un outil tel que Windows par exemple.

Un détail qui peut avoir son importance est la possibilité de prendre connaissance des différentes structures existant dans la base de connaissances, telles que la liste des règles, des hypothèses, des données, des objets, des classes ou des propriétés.

Les "réseaux" (représentations interactives graphiques du plan d'inférence et de l'univers objet) sont des fonctions de vérification et peuvent être utiles lors de recherche d'incohérences dans les inférences et dans le phénomène d'héritage.

Les éditeurs de Nexpert-Object sont conçus de façon à optimiser la tâche de construction de règles; ils sont basés sur un compilateur incrémental qui fournit un environnement unique, puissant et hautement interactif de "design" de connaissances.

Comme nous l'avons mentionné ci-dessus, l'utilisation de bases de données permet donc de séparer les connaissances et les faits stockés respectivement dans des bases de connaissances et des fichiers externes où les données peuvent être écrites ou lues pendant une consultation.

Ces bases de données couvrent trois groupes ou "modèles":

- tableurs (WKS, NXP, Excel, Lotus 1-2-3). Le contenu d'une cellule peut être associé à un donnée Nexpert ou 'object slot' (relation 1-1) ;
- bases de données (dBase III, NXPDB,...). Les informations sont stockées dans une table qui est un ensemble d'enregistrements. Par conséquent, nous avons les relations suivantes:

enregistrement <—————> objet
(une ligne de la table)

champ <—————> propriété
(une colonne de la table)

cellule <—————> 'object slot'

- bases de données relationnelles (SQL, RDB) : une structure identique à ci-dessus est obtenue (où "table" correspond à "vue" en terme relationnel).

Les interactions entre Nexpert et les bases de données sont très dynamiques et flexibles. Cette flexibilité apparaît à différents niveaux :

- les transactions peuvent être paramétrées (les critères utilisés pour sélectionner des enregistrements peuvent être basés sur des valeurs calculées auparavant par le système d'inférence) ;
- les transactions peuvent créer des objets dynamiques qui ne sont pas prédéfinis dans la base de connaissances ; ceci est fondamental pour une bonne séparation faits/ connaissances.

L'analyse que nous venons de réaliser ne recouvre pas tous les aspects qui caractérisent Nexpert-Object. N'ayant pas manipulé cet outil, il nous est impossible de prendre en compte, et surtout de connaître toutes les subtilités qu'il met à la disposition du concepteur de systèmes experts. Nous avons donc essayé d'en dégager les principales.

Passons maintenant à une analyse semblable de Guru, outil pour lequel nous pouvons faire la même remarque.

c. Guru

Guru, outil à base de règles simples, est écrit en C et en assembleur. Lancé sur le marché par Micro Data Base Systems dès 1985, il est spécialement conçu pour le monde de affaires.

Les possibilités de cet outil tournant sur PC's et hardware UNIX sont multiples. Il combine en un seul programme un shell et des outils habituels que l'on trouve dans un logiciel intégré, et notamment un système de gestion de base de données relationnelles, un tableur, des possibilités de communications avec des programmes ou fichiers externes, d'affichage graphiques, d'analyses statistiques, un traitement de textes, un système de gestion de formes et de rapports élaborés. De plus, si cela s'avère nécessaire, le shell permet d'aller chercher des données dans une base localisée dans un autre site géographiquement éloigné, via le module de télécommunications.

Sur Guru, les faits (ou variables) sont représentés par des triplets (attribut-valeur-CF) ; ils sont intégrés dans des règles qui sont manipulées suivant un processus de chaînage arrière (par défaut), ou en chaînage avant ou mixte si le développeur le définit via diverses commandes.

Les variables Guru sont de quatre types (P.Harmon, R.Maus, W.Morrissey):

- variable de travail : peut contenir des nombres, des strings, des expressions logiques, ou une valeur 'unknown' ;
- cellule : variable qui fait référence à un tableur chargé en mémoire suite à la communication 'LOAD FILE' ;
- champ : variable dont la valeur est issue de l'enregistrement courant de la base de données examinée ;
- variable d'environnement : variables qui définissent l'environnement de travail (formules de calcul des facteurs de confiance, stratégie d'inférence, couleur de l'écran, longueur maximale des messages à afficher,...).

Ces variables peuvent être réinitialisées au début de chaque consultation.

Les règles en Guru sont de forme

'NAME : IF <condition> THEN <conclusion>'

où la (ou les) conditions est (sont) une (des) expression(s) logique(s) (variable, fonctions, opérateurs logiques, arithmétiques ou relationnels),

la conclusion correspond à n'importe quelle commande Guru (assignation, input, output, calculs,...).

Outre ces nom, prémisses et conclusion, plusieurs autres options peuvent être définies :

- priority : entier de 1 à 100 qui définit un ordre dans lequel le système doit considérer toutes les règles susceptibles d'être déclenchées (s'il en existe plusieurs). Celle qui possède la valeur de priorité la plus élevée sera tirée en premier. La valeur par défaut est de 50 ;
- cost : entier de 1 à 100 indiquant le coût de "tir" d'une règle. Cette option peut être combinée avec 'priority', par exemple, pour déterminer l'ordre dans lequel deux règles de priorité identique seront déclenchées.

La valeur par défaut est de 50.

- test : lettre qui indique quelle stratégie Guru utilise lorsqu'il essaie d'évaluer la condition d'une règle. Cette option est définie localement et est donc spécifique à une variable (par opposition aux variables d'environnement qui ont quant à elles une portée globale) ;
- Précisons que lorsque Guru cherche à établir une valeur pour une variable, il considère les règles candidates dans un certain ordre. Par défaut, il prendra ces règles dans l'ordre de déclaration ; il est cependant possible de lui commander de considérer d'abord les règles de plus grande priorité, de moindre coût,... ou de les prendre au hasard (P.Harmon, R.Maus, W.Morrissey) ;

- comment : texte expliquant ce que fait la règle (ceci est plutôt destiné au développeur) ;
- reason est un texte expliquant ce que signifie la règle (plutôt destiné à l'utilisateur) ;
- changes indique la (les) variable(s) dont la (leur) valeur a été changée suite à une action ;
- ready permet de spécifier une séquence de commandes Guru ; elle sera exécutée avant que les prémisses de la règle soient évaluées.
Ce peut être la consultation de la base de données, l'extraction de données, une question à l'utilisateur, ... ;
- needs : précise les variables dont la règle a besoin pour évaluer la condition.

L'utilisateur définit lui-même s'il souhaite la prise en compte de ces options par le système. De plus, 'Changes' et 'Needs' ne sont pas nécessaires dans le cas où aucune macro-variable n'est utilisée (une macro-variable étant une variable qui contient une expression logique).

A chaque valeur d'une variable - y compris les variables floues - est associée un facteur de confiance (CF) compris entre 0 (valeur inconnue) et 100 (valeur vraie). La valeur par défaut est de 100.

Pour qu'une variable soit "connue", il faut et il suffit que son CF soit supérieur à un seuil qui par défaut vaut 20, mais qui peut être établi librement par le développeur.

Si le CF est inférieur à 100, Guru raisonne donc dans un environnement incertain. Il propose alors à l'utilisateur un choix de formules pour contrôler la façon dont le système utilise ces coefficients de certitude, et notamment la manière dont il calcule le CF correspondant aux composantes de la condition d'une règle, ou encore la relation entre le CF global des prémisses et le CF de la conclusion.

La façon dont le système calcule les CF's est désignée par "algèbre des CF's".

Les CF's peuvent être calculés pour les prémisses, définis pour les actions, et une combinaison est évaluée pour une affectation dans une action. Dans les cas où différentes règles sont susceptibles d'être tirées, le système combine ces CF's résultats.

Différentes algèbres sont possibles (manuel d'utilisation):

1. Lorsque le calcul porte sur une

- expression relationnelle
- expression logique
- expression numérique en chaîne
- affectation soustractive à une variable multivaluée,

quatre algèbres sont envisageables :

- minimum
- produit
- moyenne $(\min(a,b) + ((a*b)/100))/2$
- balance $((a*b)/100) * (2 - \max(a,b) * (1 - a/100) * (1 - b/100))$

2. Lorsque le calcul porte sur une

- expression logique avec des 'OU'
- affectation additive à une variable multivaluée,

quatre algèbres sont possibles :

- maximum
- somme des probabilités
- moyenne $(\max(a,b) + (a + b - a*b/100))/2$
- balance $\max(a,b) + \min(a,b) * (1 - a/100) * (1 - b/100)$.

Bien sûr, les utilisateurs que la précision et la perfection ne concernent pas (ou n'intéressent pas !), peuvent se reposer sur les formules déterminées par défaut.

Le choix peut se réaliser de façon globale (au moyen de variables d'environnement) ou localement (dans la définition d'une variable). Chaque variable peut avoir son propre mode de calcul de CF ; lorsque rien n'est spécifié, la stratégie "globale" est d'application.

Mais certains problèmes requièrent l'emploi de plusieurs formules durant une consultation ; dans ce cas, la variable 'E.CFVA' peut être placée dans les règles utilisant une méthode particulière de calcul de CF.

La base de connaissances, ou base de règles dans la nomenclature Guru, peut être construite de deux façons : via un éditeur spécialisé (hiérarchie de menus et fenêtres dans lesquels le développeur introduit des informations relatives aux règles, variables, but,...), ou via un éditeur de textes interne ou externe (P.Harmon, R.Maus, W.Morrissey). Puisqu'une base de règles peut en consulter une autre, des problèmes plus complexes peuvent être décomposés en éléments modulaires (manuel d'utilisation).

Il est possible de définir des variables dans des règles pour lesquelles le développeur souhaite un comportement particulier ; ainsi des informations telles que la formule de calcul des CF's à employer, la description de la variable, ou encore une clause 'FIND' indiquant comment éventuellement obtenir la valeur d'une variable inconnue, peuvent être introduites.

A noter que contrairement à VP-Expert, et dans le cas où aucune clause 'FIND' n'existe, le système Guru ne génère pas automatiquement de questions pour demander la valeur d'une variable (autoquery) (manuel d'utilisation).

Voyons maintenant les caractéristiques offertes par l'interface développeur.

Il est également très facile d'insérer des règles dans la base de connaissances, mais ce mode de représentation semble parfois insuffisant pour représenter certaines connaissances.

Pour améliorer les performances, Guru autorise des appels au DOS dans le but d'exécuter des programmes externes ; il lui est aussi possible d'interagir avec des fonctions écrites en C sans passer par le DOS.

Guru permet également que, pendant la phase de mise au point, le développeur sauvegarde les tests de règles effectués, chacun

GURU Rule Set Manager			
Rule Set:replant.rss		Goal:action	
Edit Next Prior Return	Rule: R8 Priority: 20 Cost: 20 Test:	Comment Determine if the field needs replanting.	
Ready If Viable = false and (uniform = true or stand <12000) a Then pneed = "replant" Reason If there are no viable seeds remaining in the field, and there is either uniform thinning or a stand worse than 12000 plants per acre, then the field needs to be replanted.		Needs viable uniform stand problem Changes	
Control Done Esc Help ^L	Window Prev PgU Next PgD	Line Up ↑ Down ↓ Ins ^Q Del ^T Ers ^Z End End	Word Right → Left ← Character Right ^D Insert Left ^S Delete

Figure 31 :
 Ecran menu de Guru.
 [P.Harmon et al., 1988]

d'eux correspondant à une séquence différente de réponses à des interrogations. De plus, des données rendues par le système expert peuvent être mises sous forme de graphique.

Mais globalement, l'interface développeur n'apparaît pas très puissante aux yeux de P.Harmon, R.Maus et W.Morrissey. Selon ces mêmes auteurs, il n'est pas possible d'obtenir des renseignements sur le pourquoi ('WHY' et 'HOW' décrits dans Nexpert et VP-Expert) d'une conclusion ou d'une question posée. Pourtant, d'autres personnes [manuel d'utilisation] affirment le contraire actuellement : pendant le raisonnement, et même après, il est possible de demander au système pourquoi il a tiré une règle ou encore d'afficher la suite de règles qu'il a déclenchée pour arriver au résultat.

Précisons que cette fonction est surtout utile à l'utilisateur pour qui les informations fournies peuvent être plus sommaires. Etant donné que la stratégie de travail du système peut être relativement complexe (cfr options à portée globale ou locale), il est souvent difficile, voire impossible, au système d'expliquer clairement toutes les inférences effectuées, ce qui serait effectivement intéressant pour le développeur (et à la fois trop complexe pour l'utilisateur).

Des messages, questions et autres définitions de variables d'environnement peuvent être programmés dans une procédure spéciale d'INITIALISATION exécutée avant que le système ne cherche le but. Similairement, une procédure, 'COMPLETION', contient des commandes Guru à exécuter une fois que l'objectif est visé.

En ce qui concerne l'interface utilisateur, elle est ici aussi très interactive. Guru met bon nombre de menus à la disposition de l'utilisateur, et pour faciliter l'interaction, il peut également fonctionner avec une souris (voir figure 31). De plus, ce système permet, à l'aide de nombreux outils (générateur d'écrans, commandes d'entrées/sorties très

souples,...), de présenter des informations selon le format désiré. [manuel d'utilisation]

Parmi les quatre interfaces de Guru, l'utilisateur peut choisir celle qui est la plus appropriée à son problème ou à son niveau de compétence: [manuel d'utilisation]

- l'interface langage naturel : utile pour ceux qui souhaitent seulement faire des consultations, sans avoir à apprendre un langage spécifique. Actuellement, elle est uniquement disponible en anglais ;
- menus déroulants interactifs, avec toujours un écran d'aide disponible ;
- langage de commande : peut être utilisé par ceux qui veulent exploiter les informations sans utiliser les menus ;
- langage de procédure : permet aux développeurs de réaliser des applications, de créer leurs propres menus et écrans.

Une consultation peut être amorcée suivant deux processus différents :

- en sélectionnant une option de consultation au menu
- en introduisant une commande particulière - 'CONSULT FILENAME' - dans une règle ; ceci permet donc d'entamer une consultation pendant une autre.

Si on considère l'interfaçage avec le système, on observe que certains utilisateurs continuent à préférer Lotus 1-2-3 au tableur faisant partie intégrante de Guru, du moins pour certaines applications. En effet, parfois le transfert de données entre les deux programmes peut constituer un point critique.

Après avoir détaillé quelques aspects techniques, il est utile de préciser que le package Guru est essentiellement utilisé pour développer des applications dans leur intégralité, et non pas des fragments d'application qui font déjà appel à d'autres logiciels tels que Lotus 1-2-3 ou dBase III.

Selon les trois auteurs pré-cités, à savoir P.Harmon, R.Maus, et W.Morrissey, il semble que les concepteurs de Guru n'aient pas suffisamment mis l'accent sur la façon d'intégrer d'autres programmes au système expert développé. Le vocabulaire employé dans la documentation s'y rapportant est souvent trop technique et ne semble guère adapté à ceux qui abordent le sujet avec une perspective système expert (versus les programmeurs conventionnels).

Mais il ne faut cependant pas perdre de vue que la plupart des outils mis à la disposition de l'utilisateur par Guru sont "conventionnels".

Guru est disponible sur PC XT/AT avec disque dur, 640 K sont recommandés pour la mémoire.

Dans sa version réseau local, Guru permet à plusieurs utilisateurs de partager des ressources, des applications, des programmes tout en conservant l'intégrité des données.

Le tableau 6 synthétise les principaux concepts évoqués dans cette analyse.

CRITERES UTILISES	VP-EXPERT	NEXPERT	GURU
représentation des connaissances - les faits - les règles - les objets	(attribut-valeur) IF <cond> THEN <actions> (les cond. peuvent être reliées par AND ou OR) -	(attribut-valeur) IF <cond> THEN <conclusion> and DO <actions> (les cond. peuvent être reliées par AND, NOT, mais pas OR) concepts d'îles de con- naissances, de contextes méta-règles -	(attribut-valeur-cf) NAME : IF <cond> THEN <concl> (plusieurs options peuvent être utilisées: priorité, coût, test,...) concepts de variables locales et globales règles + objets représen- tent les connaissances
mécanisme(s) d'inférence	arrière avant si on le force (?)	avant, arrière ou mixte	avant, arrière ou mixte
coefficient de certitude	de 0 (val. fausse) à 100 (val. vraie) entrée dans la BC directe- ment ou via l'utilisateur en réponse à une question divers procédés de calculs en fonction de la compo- sition des règles (con- tient des AND, OR ?)	toute connaissance est soit vraie, soit fausse cependant, le 'category editor' permet de défi- nir un degré de certi- tude "artificiellement" via l'assignation de priorité (se fait direc- tement dans la BC)	de 0 (val. inconnue) à 100 (val. connue et vraie) propose jusqu'à 16 algèbres de CF
notion d'héritage	-	ascendant, ascendant, multiple concept de 'inheritance category'	-

<p>interface développeur</p> <ul style="list-style-type: none"> - création de la BC - explication du raisonnement - autre 	<p>via un éditeur interne ou externe</p> <p>via transformation d'exemples par un 'inductive frontend'</p> <p>'How' et 'Why' disponibles (clause 'Because')</p> <p>trace des inférences faites apparaît sous forme de texte ou de graphe</p> <p>instruction 'Whatif'</p> <p>2 fenêtres permettant de suivre les inférences et les résultats au cours d'une consultation</p>	<p>via un éditeur interne basé sur un compilateur incrémental</p> <p>'How' et 'Why' disponibles</p> <p>traces apparaissent sous forme de graphe</p> <p>2 fenêtres permettent de suivre les inférences et les conclusions</p>	<p>via un éditeur interne, externe ou spécialisé</p> <p>'How' et 'Why' (?)</p>
<p>interface utilisateur</p> <ul style="list-style-type: none"> - menus - graphes 	<p>très conviviaux</p>	<p>très conviviaux</p> <p>existence d'une souris qui facilite l'interaction</p> <p>possibilité d'intégrer des graphes si nécessaire</p> <p>possibilité d'obtenir la liste des structures existant dans la BC</p> <p>concept de réseau</p>	<p>nombreux</p> <p>présence d'une souris qui facilite l'interaction</p> <p>données peuvent être mises sous forme de graphiques</p> <p>l'utilisateur a le choix entre 4 interfaces</p>
<p>interface système</p> <ul style="list-style-type: none"> - accès à des fichiers externes 	<p>fonctions 'Fast' et 'Slow' présentes</p> <p>accès à de multiple records ou cellules d'une BD ou d'un tableur respectivement</p> <p>accès à des fichiers externes via un appel au DOS</p>	<p>accès aux tableurs, BD et BD relationnelles</p> <p>accès à des fichiers externes via un appel à l'interface externe ('callable interface')</p>	<p>accès à une seule cellule/record</p> <p>accès à un tableur interne ou externe</p> <p>accès à des fichiers externes via un appel au DOS</p> <p>appel à des fonctions écrites en C sans passer par le DOS</p>

software			
- langage dans lequel est écrit le logiciel	C	C - Pascal - assembleur	C - assembleur
- OS requis	DOS version 2xx, 3xx		
hardware	PC	VAX	PC XT/AT avec disque dur hardware UNIX
remarque			le shell est un des multiples outils offerts par Guru au même titre que le SGBD, le traitement de texte,...

légende : BC = base de connaissances
CF = coefficient de certitude

3.3.3. Synthèse

En conclusion, nous pouvons observer que les trois outils analysés présentent des caractéristiques avantageuses, et d'autres qui le sont moins.

Nous avons notamment examiné l'introduction de règles dans la base de connaissances, l'utilisation de facteurs de confiance, l'accès à un tableur et à une base de données, l'interfaçage avec des programmes externes, avec le concepteur et l'utilisateur,...

Fort heureusement, ce ne sont pas systématiquement les mêmes propriétés qui sont bénéfiques pour les différents shells, ni toujours les mêmes qui sont défavorables.

Ce que nous devons retenir de cette étude est qu'il n'existe pas de bons ou de mauvais outils. Simplement, il convient d'effectuer un choix optimal en fonction de trois concepts clés :

- la nature du problème à résoudre, et sa complexité ;
- la disponibilité de l'expert et le degré d'expertise des connaissances ;
- les caractéristiques offertes par le shell.

Nous voyons que le point crucial est de bien délimiter et cerner le problème, et ensuite de convertir, transformer l'expertise recueillie en connaissances utilisables par le système.

Les outils eux-mêmes ne sont donc pas aussi importants que l'analyse qui conduira au développement du système expert.

Avant tout, il est donc nécessaire de bien se documenter, et de ne pas se hâter à sélectionner un shell.

En ce qui concerne notre travail, étant donné nos connaissances relativement limitées quant aux shells disponibles sur le marché, notre promoteur nous a conseillé l'usage de VP-Expert. De plus, quelques shells "seulement" sont disponibles officiellement à l'Institut d'Informatique.

Comme nous l'avons précisé, les fonctionnalités présentes sur cet outil semblent proportionnelles à l'ampleur du problème à résoudre et à sa nature. De plus, la dimension heuristique des connaissances que notre système manipule est faible, et permet donc l'usage de VP-Expert.

Nous avons toutefois éprouvé quelques difficultés - liées aux limites intrinsèques de l'outil - lors de l'implémentation. Nous détaillerons ce point ultérieurement.

3.4. Marché, applications et exemples de systèmes experts

Intéressons-nous tout d'abord au marché des systèmes développés en intelligence artificielle.

Les compagnies américaines ont maintenant accepté les idées clés en intelligence artificielle et sont passés du stade de la recherche expérimentale à celui de l'implémentation des systèmes experts. Ayant découvert les bénéfices de l'utilisation des bases de connaissances dans la résolution d'une grande variété de problèmes, elles sont prêtes à commercialiser ces nouvelles technologies dès qu'elles auront quitté les laboratoires de recherche.

Durant les trois dernières années, alors que la plupart des ventes informatiques ont chuté, les ventes de systèmes experts se sont révélées en constante progression. De nombreuses entreprises croient maintenant que les systèmes experts vont leur fournir le lien nécessaire permettant d'intégrer les personnes et les technologies informatiques dans des organisations plus efficaces.

D'autre part, l'introduction de différents outils de développement de systèmes experts fonctionnant sur des IBM PC's a significativement augmenté le nombre d'individus et de compagnies pouvant débiter l'expérimentation de systèmes experts. Au plus les gens vont apprendre les fonctions de tels systèmes, au plus la demande ira vers des outils plus sophistiqués.

On admet que l'expansion des systèmes experts sera foudroyante dans les cinq prochaines années. Dans les années 1990, ils auront acquis un usage général, exécuteront une multitude d'applications dans un grand nombre de domaines et, enfin, la plupart des compagnies opteront définitivement pour cette nouvelle technologie et, ce simplement dans l'objectif de rester compétitives.

Venons-en maintenant aux différentes applications des systèmes experts. Nous donnons ci-dessous une liste non limitative de tels domaines avec pour chacun les problèmes que l'on cherche à résoudre par le développement de systèmes experts :

- aéronautique et espace : aide au pilotage, planification du trafic aérien, contrôle des satellites,...
- agriculture : diagnostic de maladies et conseils thérapeutiques en pathologie végétale, aide à la gestion d'une ferme,...
- banque, finance et assurance : gestion financière, audit, aide au placement,...
- biologie : de très nombreux projets de systèmes experts ont vu le jour dans ce domaine. Ils concernent :
 - * l'aide au diagnostic et à la décision, activités très proches de celles rencontrées en médecine,
 - * la recherche clinique,
 - * la planification d'expériences de laboratoire ; par exemple MOLGEN, un des premiers systèmes experts, destiné à aider le généticien dans la planification d'expériences de clonage en biologie moléculaire,
 - * l'interprétation de données physico-chimiques, notamment pour l'élucidation de structures,
 - * l'interprétation d'images de microscopie, radiographie,...
- chimie : aide à la synthèse de nouvelles molécules organiques, élucidation de la structure tridimensionnelle de molécules,...
- électronique : diagnostic de pannes de circuits électroniques, de centraux et de réseaux téléphoniques,...
- formation et enseignement
- géologie : aide à la prospection géologique et minière, gestion de forages,...
- industrie : conduite de processus industriels, CAO, gestion de production,...
- informatique : aide à la programmation, configuration de systèmes informatiques, aide à l'exploitation et à la maintenance,...

- médecine : le domaine médical a donné lieu dès le début à de multiples expériences, même si les systèmes réellement opérationnels sont peu nombreux. En revanche, on peut noter une utilisation importante des systèmes experts comme aide à l'enseignement pour des étudiants en médecine. Les applications concernent :

- * l'aide au diagnostic : de nombreux domaines ont été abordés à la suite de MYCIN (diabète, ictères, glaucomes, poumons, maladies cardio-vasculaires),
- * la surveillance de malades sous soins intensifs,
- * l'aide à l'interprétation d'images et signaux médicaux,...

Parmi les nombreux systèmes experts existant sur le marché, nous vous en présentons trois qui figurent parmi les plus courants.

3.4.1. MYCIN

Il s'agit d'un système développé vers le milieu des années 1970 à l'Université de Stanford. Il a été le premier grand système expert à travailler au niveau d'un expert humain, et à fournir aux utilisateurs une explication de son raisonnement. Ce système a été élaboré pour aider les médecins dans le diagnostic et le traitement des infections bactériennes du sang et des méningites. Ces maladies pouvant être fatales très rapidement, le médecin doit commencer un traitement en l'absence des résultats complets du laboratoire ; c'est pourquoi le médecin consulté cherche souvent à avoir les conseils d'un expert.

MYCIN pose à l'utilisateur des questions sur le patient, et fournit un ensemble ordonné de diagnostics possibles et de recommandations de thérapies. Divers projets ont été réalisés autour de MYCIN. Notamment, Guidon est un système

expert d'enseignement assisté par ordinateur. En utilisant la connaissance de MYCIN, il enseigne à l'étudiant les données cliniques et de laboratoire intéressantes à envisager, et explique comment utiliser cette information pour diagnostiquer l'organisme responsable de l'infection.

3.4.2. XCON

XCON, initialement appelé R1, date de la fin des années 1970. Il s'agit d'un système de design opérationnel qui configure les ordinateurs de la famille VAX pour Digital Equipment Corporation (DEC).

Ce système reçoit une commande d'un client et fournit un ensemble de diagrammes présentant les relations spatiales entre les composants. Ces diagrammes sont utilisés par les techniciens qui assemblent physiquement le système. XCON possède quelques 6000 règles au sujet des différentes manières d'assembler les parties constitutives d'un ordinateur. Il est effectivement utilisé depuis 1981 et il est à noter qu'il s'avère parfaitement rentable.

3.4.3. DENDRAL

Le projet DENDRAL fut développé par Buchanan, Mitchell et Feigenbaum vers la fin des années 1960. Sa tâche consiste en l'analyse de données provenant d'un spectrographe de masse, et en la détermination des structures moléculaires qui donnent probablement lieu à ces spectres.

De par ses performances considérables, DENDRAL est très largement utilisé par de nombreux chimistes américains. Ce système a démontré l'utilité des méthodes et des outils d'intelligence artificielle dans un véritable environnement scientifique, et a focalisé l'attention sur les issues techniques et sociales dans le domaine du "knowledge engineering".

III. PRESENTATION DU TRAVAIL

III. PRESENTATION DU TRAVAIL

1. Aperçu du travail réalisé antérieurement

Comme il a été dit précédemment, ce travail constitue un prolongement à un mémoire de fin d'études réalisé l'année dernière par deux étudiants en Informatique. Nous allons résumer brièvement leurs objectifs ainsi que leurs aboutissements.

Notons tout d'abord que leur objectif premier était de montrer la faisabilité du développement d'un système expert dans le domaine de la biochimie cellulaire. Pour ce faire, ils ont consacré une grande partie de leur temps à l'élaboration d'une analyse d'opportunité, afin d'étudier les multiples facettes de la biochimie susceptibles d'être implémentées en terme d'intelligence artificielle. Parmi celles-ci, la purification de protéines s'est révélée être celle qui se prêtait le mieux à ce genre de modélisation. En effet, la conduite d'un plan de purification par un outil de type système expert peut constituer un apport pour le laboratoire. D'abord, le processus d'acquisition des connaissances auprès des experts leur permet de structurer et de formaliser ces dernières. Ensuite, le système expert permettra éventuellement d'éduquer des non-spécialistes (étudiants, opérateurs,...).

Eu égard à ces objectifs, la tâche de ces deux étudiants a consisté, dans un premier temps, en une immersion dans l'univers biologique (et plus particulièrement biochimique) afin d'acquérir les connaissances nécessaires à la réalisation d'un tel système, mais surtout afin de comprendre, percevoir les

concepts fondamentaux de cette discipline qui leur était étrangère.

Dans un second temps, ils ont décidé la prise en charge par le prototype de système expert de trois techniques de purification de protéines, dont deux furent implémentées (à savoir la précipitation isoélectrique et la chromatographie sur échangeur d'ions), les règles concernant la troisième technique (le tamis moléculaire) ayant été établies, mais non intégrées au système.

Leur travail a donc principalement consisté en la réalisation d'un "prototype noyau" susceptible d'être ultérieurement agrémenté de mécanismes le rendant plus performant et convivial.

Il faut cependant noter que ce système a été réalisé en Lisp et que par conséquent, le développement du moteur d'inférence a considérablement accru la difficulté de la tâche.

Après ce considérable travail réalisé et la prise de conscience qu'un tel système expert pouvait s'avérer fort utile dans le domaine biochimique, il fut convenu de donner une suite à ce mémoire et ce, dans le but principal de réaliser un système qui soit réellement opérationnel sur des cas réels. Des objectifs secondaires consistent en un approfondissement des connaissances de l'expert, en un ajout de certaines techniques, mais surtout en l'élaboration d'un plan de purification (c'est-à-dire la nécessité de définir des critères précis d'enchaînement des différentes techniques implémentées). Une interface plus agréable et un mécanisme d'explication du raisonnement contribueraient à améliorer le travail déjà effectué.

Afin de tendre vers ces objectifs, nous nous sommes aidées non plus d'un langage de programmation (Lisp), mais bien d'un outil de développement de système expert (VP-Expert).

Nous exposons ci-dessous notre approche du problème ainsi que les phases principales de la méthode d'investigation suivie.

2. Méthode d'investigation

2.1. Identification du problème et des acteurs

Comme il a été décrit au chapitre II, le développement d'un système expert débute par une phase d'identification la plus précise possible du problème et des concepts qui y sont associés, des tâches que doit prendre en charge le système, ainsi que des objectifs visés par le concepteur.

Dans cette section, nous nous attacherons à présenter ces différents aspects.

2.1.1. Identification du problème et de l'environnement de travail

En parcourant les chapitres précédents, le lecteur perspicace aura certainement compris que notre travail concerne la purification de substances d'origine biologique, et plus particulièrement la purification de protéines.

On peut se poser la question de savoir pourquoi les chercheurs ont recours à de telles techniques. En réalité, une protéine issue d'un corps vivant (sang, tissu,...) n'est pas à l'état pur, et donc ne peut être manipulée comme telle par le biochimiste. En effet, d'autres substances présentes dans l'échantillon recueilli pourraient nuire au bon déroulement des

expériences ou affecter la validité des résultats obtenus. Il est donc important de pouvoir séparer les contaminants de la protéine faisant l'objet de l'étude.

Pour réaliser correctement cette séparation, le scientifique doit se baser sur un plan de purification, c'est-à-dire une suite de techniques permettant d'isoler une substance particulière contenue dans un mélange complexe.

Ce mélange peut contenir trois classes de substances :

- la protéine à purifier,
- les contaminants empêchant la manipulation de la protéine dans l'état actuel de l'échantillon, et qui donc doivent être complètement éliminés
- les substances contaminantes qui n'affectent pas la protéine, et qui donc peuvent "cohabiter".

Le problème n'est donc pas aisé à résoudre. La résolution requiert une certaine expertise ayant trait aux techniques applicables dans une telle situation (cfr chapitre I). Dans cette approche, nous nous sommes limitées à trois techniques de séparation afin de circonscrire le problème. Il s'agit de la chromatographie sur échangeur d'ions, le tamis moléculaire, et l'HPLC. D'autres techniques, la précipitation isoélectrique, l'électrophorèse et l'électrofocalisation, n'ont pu être abordées que sommairement en raison de l'impossibilité de déterminer des critères d'application clairs et précis. En effet, il est souvent difficile, parfois imprécis, voire erroné, de reproduire et de réduire des éléments de la nature - qui est pour une grande part la réalité scientifique - dans des modèles valides en vue de les exploiter. Il est ardu, et à la fois délicat, de vouloir définir des seuils formels et des normes quantifiables pour des phénomènes qui, en fait, ne sont que subtilité et nuance.

En conséquence, il est dangereux pour le biochimiste de considérer aveuglément les conseils prodigués par le système.

Précisons encore que la pratique d'une technique est indispensable pour en acquérir une connaissance approfondie. Certaines techniques sont effectivement inadéquates pour un mélange particulier, soit parce qu'elle est inefficace et ne permet pas une séparation suffisante de la protéine à purifier, soit parce qu'elle dénature le matériel à analyser.

Il est également impératif que le biochimiste dispose d'informations propres à l'échantillon, et aux éléments qui le composent. L'utilisation des techniques qui sont à sa disposition en sera d'autant plus efficace, parce que mieux adaptée au mélange en question.

Cependant, les connaissances que possède le chercheur sont souvent imparfaites et partielles, à moins qu'il ne s'agisse d'un échantillon très simple. Dès lors, sur base de ces informations, il est très difficile au système de guider le scientifique dans son investigation.

De plus, les connaissances sont souvent réparties entre les biochimistes.

Les recherches dans ce domaine étant longues et difficiles, le chercheur ne base pas toujours la construction d'un plan de purification sur des connaissances précises. Il adopte parfois un travail par "essais et erreurs" sur des petits échantillons du mélange. Certaines séparations efficaces peuvent ainsi être mises au point sans que l'on puisse apporter une explication claire des raisons du comportement des composants du mélange, ou sans qu'on connaisse véritablement les substances éliminées.

Nous constatons que l'ingrédient essentiel d'un système expert est la connaissance du domaine. Il est indispensable que le système appuie son raisonnement sur des informations aussi vastes que possible, tout en restant claires et bien définies.

Bien entendu, pour construire un plan de purification, il faut joindre à la connaissance des techniques et des mélanges, la capacité de choisir et d'ordonner judicieusement ces techniques, afin que la séparation soit la meilleure et la plus rapide possible, tout en entraînant un minimum de perte. En effet, certaines techniques ont recours à des éléments qui sont incompatibles avec des réactifs utilisés dans d'autres techniques - il en va ici de la non dénaturation de la protéine à isoler.

Il convient donc de prendre ces contraintes en compte et d'en avertir le biochimiste le cas échéant.

Nous en parlerons d'ailleurs plus en détails au cours de la section 2.3.3..

Le point de départ pour la construction d'un plan de purification repose donc sur les informations concernant les substances du mélange. Puisque le système ne dispose pas en général d'une description complète, il construit le plan dans un univers incertain.

Plus les informations seront précises, meilleure sera la construction.

Une fois développé, le système doit être capable d'aider l'utilisateur à construire un plan de purification efficace pour un échantillon quelconque. Il conseille l'utilisateur sur la suite de techniques à appliquer à un échantillon préalablement décrit, pour séparer la substance d'origine biologique à purifier des substances qui la contaminent.

Le plan doit être construit en constante interaction avec l'utilisateur. Celui-ci reçoit le conseil du système et prend une décision. Le système travaille toujours en fonction de la réaction de l'utilisateur.

Non seulement le système doit être capable de conseiller le biochimiste sur la suite adéquate de techniques à appliquer à un échantillon, mais il doit également lui indiquer une évaluation du résultat obtenu suite à l'application de ces techniques afin que le chercheur puisse juger de l'applicabilité du plan.

Ce plan comporte donc différentes étapes. Pour chacune d'elles, le système présente les conditions d'application de la technique (matériel de laboratoire, conditions expérimentales, ...), et les résultats auxquels le chercheur peut s'attendre (durée de la technique, perte par dénaturation du matériel biologique, ...).

La qualité du conseil donné par le système dépend de la qualité et de la quantité des connaissances dont il dispose. Puisque ces connaissances sont rares, la mise au point d'un système ayant une efficacité réelle risque d'être longue. De plus, elle doit notamment permettre d'alimenter la base de connaissances du système. Il faut également ajouter que, même si le système ne dispose pas de beaucoup d'informations, il doit être capable de fournir un conseil. L'utilisateur doit alors pouvoir estimer la confiance qu'il peut accorder aux conseils prodigués, et ainsi guider la construction du plan. En aucun cas, le système ne lui impose l'application d'une technique.

2.1.2. Identification des acteurs

Le laboratoire de biochimie cellulaire avec lequel nous avons collaboré est équipé en matériels informatiques (notamment des Macintosh, des PC's reliés à des appareils de mesure du laboratoire) ; le personnel y recourt largement dans le cadre de son travail, et ceci a certainement facilité le contact. En effet, les membres du laboratoire sont en général très ouverts à toute explication ou information relatives à notre travail ; leur intérêt pour ce domaine les a amenés à émettre des critiques sur notre façon de concevoir le problème, et ceci nous a finalement été bénéfique.

En retour, nous leur avons fait part de l'avancement de notre projet ; cependant, nous ne les avons jamais entretenus au sujet des problèmes purement informatiques auxquels nous étions confrontées.

Ajoutons que nous avons principalement travaillé avec deux personnes du laboratoire avec qui nous sommes restées en "interaction" pendant toute la durée de notre mémoire.

2.1.3. Spécificités d'un système expert de planification

Au cours du chapitre II, nous avons décrit les différents types de systèmes experts qu'il était possible de développer. Au vu de cette classification, nous pouvons dire que notre prototype s'inscrit dans le groupe des systèmes experts de planification, c'est-à-dire que c'est un programme qui, à partir d'un état initial et d'un état final, conçoit un plan - qui est une suite d'actions à effectuer pour atteindre un objectif - permettant de passer de l'une à l'autre.

La première caractéristique importante d'un système expert de ce type est l'état initial de l'objet. C'est sur cet objet que débute le travail du système. Cet état initial est décrit par l'utilisateur du système, cette description n'étant pas nécessairement exempte de lacunes ou d'imprécisions. Elle se compose d'un certain nombre de caractéristiques prédéfinies par le concepteur du système, et dans le cas qui nous préoccupe, l'état initial décrit l'état de l'échantillon avant toute purification. Certaines caractéristiques sont contenues dans la base de connaissances du système, les autres seront fournies par l'utilisateur. Mais il est extrêmement rare que toutes les informations relatives à chacune des protéines du mélange soient connues.

Une seconde caractéristique est l'état final de l'objet. Il est assimilable à un état optimal au sens où il s'agit de l'état souhaité par l'utilisateur, ou du moins le plus proche de cet état, en fonction des connaissances du système. L'état final est donc celui où devra se trouver l'objet après que la suite d'actions conseillées par le système ait été exécutée.

Dans l'exemple qui nous concerne, l'état final spécifie l'état de l'échantillon après purification. Le système considère qu'il a atteint cet état lorsque l'échantillon est complètement purifié, en l'occurrence quand l'utilisateur lui signale que la purification obtenue lui semble suffisante.

Une autre caractéristique d'un système de planification est le plan conseillé par le système. Il est inféré à partir des états initial et final de l'objet. Le système utilise pour ce faire des informations emmagasinées dans une base de connaissances. Elles ont été fournies par un ou plusieurs expert(s) dans le domaine spécifique du système.

La dernière caractéristique importante d'un système de planification est la manière interactive et incrémentale selon laquelle il travaille le plus souvent. Le système formule, raffine, et tente de résoudre un problème décrit par l'état de l'objet via une "conversation" avec l'utilisateur. Au cours de cette interaction, le système propose à l'utilisateur des solutions partielles possibles en fonction de la formulation courante du problème. Il conçoit donc son plan de manière incrémentale.

Il est à noter que cette dernière caractéristique n'est pas l'apanage des systèmes de planification. On la retrouve notamment, par exemple, dans les systèmes de "design".

2.1.4. Objectifs du système

La réalisation de la tâche par un outil de type système expert peut constituer un apport pour un laboratoire. D'abord, l'acquisition des connaissances auprès des experts est, pour eux, une "épreuve" intéressante. Ils doivent structurer leurs connaissances et formaliser un ensemble de pratiques informelles.

Ensuite le système doit permettre une distribution de l'expertise auprès de non-spécialistes. Nous pensons plus particulièrement aux opérateurs et aux étudiants du laboratoire. Le système peut également apporter une aide au niveau de la réflexion sur l'enchaînement des techniques au sein d'un plan de purification.

2.2. Conceptualisation

Au cours de la phase de conceptualisation, nous présenterons les concepts et relations ayant trait au problème, et nous décrirons les différentes tâches que le prototype doit prendre en charge.

2.2.1. Concepts et relations

Les substances biochimiques composant les échantillons sur lesquels travaille le système peuvent être d'origine biologiques (comme les protéines) ou chimiques (comme les sels). Les substances auxquelles nous nous intéressons sont d'origine biologique.

Afin de déterminer les techniques de purification à appliquer à un échantillon, nous nous sommes basées sur différentes caractéristiques de ces substances :

- le nom,
- la nature (qui spécifie l'origine : protéine, enzyme,...),
- le poids moléculaire,
- le point isoélectrique,
- le domaine de stabilité,
- le (ou les) domaine(s) de précipitation,
- la stabilité dans le temps.

Nous avons adopté le terme de mélange pour désigner tout extrait de tissu susceptible d'être manipulé lors d'une purification. Ses caractéristiques concernent essentiellement l'identification des substances qui le composent :

- nom de la substance,
- sa proportion dans le mélange.

L'échantillon constitue la partie d'un mélange qui est purifiée. Précisons que nous n'avons pas toujours fait la distinction entre ces deux termes au cours de la rédaction du mémoire et du programme.

Les caractéristiques d'un échantillon sont :

- le volume,
- la quantité totale de substance,
- la concentration,
- le nom de la protéine à purifiée,
- les substances contaminantes.

Elles sont importantes surtout pour déterminer s'il est nécessaire de "préparer" l'échantillon avant l'application d'une technique de purification particulière.

Les techniques de purification - qui ont largement été développées au chapitre I - sont appliquées aux échantillons pour obtenir une séparation.

La relation "s'applique à" est définie si, et seulement si, les conditions d'application d'une technique sont compatibles avec les caractéristiques de l'échantillon.

La réalisation de cette relation fournit un nouvel échantillon. Ajoutons à cela que d'autres relations peuvent également être définies : le mélange "contient" des substances, et un échantillon "est une occurrence d'" un mélange.

Nous ne nous y attarderons pas dans le cadre de ce travail.

2.2.2. Tâches

La première tâche que le système poursuit est la détermination en quelque sorte de l'échantillon à purifier. Au cours des différents travaux qui lui seront fournis, le système acquerra des connaissances sur les substances ; une fois acquises, celles-là pourront être utilisées.

Après avoir présenté à l'utilisateur un état de ses connaissances, le système l'interroge dans le but d'enrichir cet acquis. Ainsi, l'utilisateur a la possibilité de corriger une information incorrecte ou d'entrer de nouvelles connaissances, et puis de les compléter.

Voici en quelques lignes comment il procède :
il tente d'obtenir le plus grand nombre d'informations possible relatives à l'échantillon, et pour cela, il demande :

- à l'utilisateur si la protéine à purifier se trouve dans le catalogue des éléments stockés dans la base de données qui s'affiche à l'écran
 - si oui : le système affiche ses caractéristiques (nature, poids moléculaire (PM), point isoélectrique (PI),...)
 - sinon : il demande à l'utilisateur d'introduire le nom de la protéine , ainsi que la valeur de toutes les autres caractéristiques

- d'introduire le volume de l'échantillon, sa masse et son origine
 - si volume et quantité sont trop importants, ou que c'est une protéine membranaire :
 - il est nécessaire de "préparer" l'échantillon pour qu'une technique de purification plus performante soit applicable, et en l'occurrence, il faut effectuer une précipitation isoélectrique
 - si le volume est inférieur à un seuil et que c'est une protéine soluble : le système poursuit son investigation
- si la valeur du PI pour tous les contaminants est connue
 - si oui : l'utilisateur les introduit
 - sinon : le système conseille une électrofocalisation afin de les déterminer
- si la valeur du PM pour tous les contaminants est connue
 - si oui : l'utilisateur les introduit
 - sinon : le système conseille une HPLC afin de les déterminer...

Une fois que ces différents renseignements sont pris, le système possède une description plus ou moins précise de l'échantillon de départ qui va lui permettre dans un second temps de guider le biochimiste dans son travail. Il le fera par la construction d'un plan de purification de façon interactive.

Partant de cette description, le processus itératif propose un choix de techniques de purification applicables à cet échantillon. L'utilisateur peut alors sélectionner la technique qui lui semble la plus adéquate, et observer l'évaluation que le système lui propose.

Bien sûr, si les propositions avancées par le système ne lui conviennent pas, il lui est loisible de modifier son choix, et d'opter pour une autre technique.

Lorsque le système présente à l'utilisateur les techniques qu'il est possible d'appliquer, il le fait en se basant sur une série de critères.

Il s'agit effectivement de vérifier si les critères de choix d'une technique sont compatibles avec les caractéristiques de l'échantillon, et si les propriétés sont telles que l'application de la technique permette de progresser dans la purification.

Ces critères seront exposés en détails au point 2.2.3..

De plus, il ne faut pas envisager les techniques dans un ordre quelconque. L'application de certaines d'entre elles au début d'un plan est plus avantageuse, alors que l'application de ces mêmes techniques en fin de plan l'est nettement moins. Par exemple, la précipitation isoélectrique est plutôt une technique appliquée au début d'un plan.

L'évaluation de la technique consiste à déterminer le temps nécessaire à sa réalisation, et les pertes qu'elle occasionne.

Cette évaluation est en fait établie en fonction des informations dont le système dispose. L'exécution du programme correspond en quelque sorte à une simulation.

De plus, il faut "construire" la description du nouvel échantillon à purifier (pouvoir estimer le volume résultat par exemple).

Enfin, le système présente un bilan de l'application de la technique à l'utilisateur : le matériel de laboratoire nécessaire à cette application, sa durée, la perte par dénaturation estimée pour cette technique et globalement pour toutes les techniques réalisées jusque là (pertes cumulées). C'est sur base de telles informations que l'utilisateur peut signifier au système de continuer le travail ou d'arrêter la construction, la purification obtenue le satisfaisant.

2.2.3. Critères pour déterminer l'application d'une technique

En vue d'orienter l'utilisateur vers une technique particulière, différents points peuvent être pris en compte :

- critères pour le choix du support (colonne, gel, tampon),
- critères pour le choix de la technique proprement dite,
- critères pour l'évaluation de la technique (volume résultat, durée, perte d'activité).

a. La précipitation

a.1. Intérêt de la précipitation

La précipitation, utilisée précocement dans les étapes de purification, permet de laisser dans le surnageant beaucoup de petites molécules tout en concentrant l'échantillon.

a.2. Supports d'une précipitation

Pour réaliser cette technique, l'opérateur doit disposer d'un récipient contenant le mélange à purifier et d'un agent de précipitation. On utilise habituellement une solution de sulfate ammonique saturée à 4°C et de pH 7.4.

a.3. Critères pour le choix de l'application de la précipitation

Nous avons opté pour deux critères permettant d'orienter le programme vers une précipitation. Le premier se réfère au **volume de l'échantillon**. Au-dessus de 18 ml, nous conseillons à l'utilisateur la réalisation d'une précipitation, et ce afin de réduire ce trop grand volume. Le second critère a trait à la texture de l'échantillon. Si la texture du mélange est **membranaire**, il faut séparer les protéines de membrane des autres protéines du mélange et, par conséquent, appliquer une précipitation. Par contre, si la texture du mélange est soluble, cette technique ne doit pas être envisagée.

Après la réalisation d'une précipitation, nous demandons à l'utilisateur d'entrer le nouveau volume de l'échantillon ainsi que la nouvelle quantité, et ce, afin de pouvoir calculer la concentration du mélange.

a.4. Evaluation de l'application de la précipitation

1°) Evaluation du volume de l'échantillon recueilli

Il faut connaître la quantité de matériel qui a précipité pour le pourcentage de solution saturée utilisée.

- Si l'on envisage un échangeur d'ions, il n'y a pas de limite de volume (proportion de 10 ml pour 100 mg de matériel),
- Si l'on envisage un tamis moléculaire, le volume est limité à 20 ml.

2°) Evaluation de la durée de la précipitation

La durée approximative de cette technique est de 1 heure.

b. La chromatographie sur échangeur d'ions

b.1. Intérêt de la chromatographie sur échangeur d'ions

La chromatographie d'échange d'ions, classée dans la série des chromatographies d'adsorption, possède entre autres l'intéressante propriété de concentrer l'échantillon et est ainsi souvent utilisée dans les premiers stades de purification. La facilité de mise en oeuvre et la disponibilité de nombreux supports de nature chimique variée font que la chromatographie sur échangeur d'ions est très répandue.

b.2. Supports d'une chromatographie sur échangeur d'ions

Pour réaliser un échangeur d'ions, l'opérateur de laboratoire doit disposer, notamment, d'une colonne, d'un gel et d'un tampon. L'objectif du système est de réaliser les choix les plus adéquats parmi les supports connus et utilisés au laboratoire.

1°) La colonne

La colonne est caractérisée par sa hauteur et son diamètre. Le système ne doit retenir aucune caractéristique concernant les colonnes utilisées pour une chromatographie sur échangeur d'ions.

2°) Le gel

Le système doit mémoriser le type, la marque et la charge électrique de chaque gel. Nous avons tenu compte des trois gels les plus fréquemment utilisés au laboratoire. Leurs caractéristiques se retrouvent dans le tableau 7 ci-dessous :

type	marque	charge électrique
CM	Sephadex C50	négative
SP	Sephadex C50	négative
DEAE	Sephadex A50	positive

Rappelons toutefois que les gels négatifs fixent les protéines chargées positivement, tandis que les gels positifs fixent les protéines chargées négativement.

3°) Le tampon

Le tableau 8 ci-dessous contient la liste des tampons habituellement utilisés et, pour chacun d'eux, leur pK :

nom	pK
acide acétique	4.75
cacodilate	6.20
citrate	3.10/4.70/5.40
glycine	9.90
phosphate	7.20
TRIS-HCl	8.20

b.3. Critère pour le choix de l'application de la chromatographie sur échangeur d'ions

Un seul critère nous a permis de déterminer si l'application d'un échangeur d'ions était possible sur un mélange de substances biologiques :

**les points isoélectriques (PI) des protéines
contaminantes doivent être suffisamment différents
du PI de la protéine à purifier.**

En effet, si deux protéines ont des points isoélectriques voisins, elles sortiront ensemble de la colonne et donc ne seront pas séparées par l'échangeur. Afin d'affiner ce critère, nous avons admis que pour séparer complètement des protéines sur un échangeur d'ions, il faut que ces dernières aient des PI différents de 1 unité au moins. Notons que contrairement à d'autres chromatographies, aucun critère de volume n'entre en ligne de compte ici.

b.4. Evaluation de l'application de la chromatographie sur échangeur d'ions

Après avoir décidé de la possibilité d'utiliser un échangeur d'ions, le système doit conseiller l'utilisateur à propos de différents choix relevant de l'application de la technique elle-même et lui indiquer l'échantillon résultant de son application.

1°) Choix du gel

La stratégie utilisée dans une chromatographie sur échangeur d'ions consiste à fixer la protéine que l'on veut purifier sans fixer les autres constituants du mélange, c'est-à-dire les contaminants de la protéine.

Avant de pouvoir déterminer l'utilisation d'un gel bien précis, il faut calculer le pH de travail. Le programme, connaissant le PI de la protéine à purifier et le domaine de stabilité (valeurs de pH où la protéine reste stable) de celle-ci, conseillera à l'utilisateur de travailler à un pH qui, tout en restant dans le domaine de stabilité, est le plus éloigné possible du PI.

Ce pH prendra donc la valeur de la borne du domaine de stabilité la plus éloignée du PI de la protéine purifier.

En ce qui concerne le choix du gel proprement dit, selon que le pH est supérieur ou inférieur au PI de la protéine, le système conseillera un gel respectivement positif (DEAE) ou négatif (CM ou SP).

Cependant, dans le cas d'un gel négatif, deux choix restent encore possibles :

- si le pH est inférieur ou égal à 5, le système conseillera le SP, qui est plus dénaturant que le CM mais qui garde sa forte charge négative dans une zone de pH bas,
- si le pH est supérieur à 5, le système optera pour le CM.

2°) Choix du volume du gel

Pour les différents gels connus du système, la quantité maximale de protéine qu'il est possible de fixer est de 3.75 grammes de protéine par gramme de gel sec. Cependant, pour la chromatographie, on garde une marge de sécurité de 3 grammes par rapport à la quantité maximale de fixation.

Dès lors, le volume du gel se calculera suivant le rapport suivant :

$$3.75 \text{ grammes de protéine} / 3 \text{ grammes de gel sec.}$$

Un gramme de gel sec correspond habituellement à un volume de 18 ml de gel gonflé. Ce volume de gonflement dépend du type de gel et de la force ionique du tampon utilisé. Dans cet objectif, on travaille généralement à des forces ioniques supérieures ou égales à 0.1, car, en-dessous de cette limite, il se produit un gonflement exagéré du gel.

Le volume du gel se calcule donc de la façon suivante :

$$\text{volume du gel} = \text{quantité déposée} * (3.75 / 3) * 18$$

3°) Choix de la colonne

Il n'existe pas de règle précise permettant de déterminer la hauteur de la colonne. L'utilisateur entre une hauteur située entre 10 et 20 cm.

Le diamètre est défini par la formule arithmétique suivante :

$$\text{diamètre} = 2 * \text{sqrt} (\text{volume du gel} / (\text{hauteur colonne} * 3.14))$$

4°) Choix du tampon

Le système conseillera un tampon de manière à ce que le pH auquel il a décidé de travailler et le PI de la protéine à purifier soient compris dans la zone tampon. Si plusieurs tampons sont envisageables, il les proposera tous.

5°) Choix du type d'élution

Pour décrocher les protéines, l'utilisateur a le choix entre modifier le pH ou modifier la force ionique μ . Il n'existe pas de critère précis pour effectuer ce choix.

La modification de la force ionique s'effectue à l'aide du NaCl. Cependant, il arrive qu'un sel plus fort soit utilisé : le NaSCl.

D'autre part, l'utilisateur peut choisir de décrocher les protéines par gradient ou par modification des conditions en étapes.

Le décrochage par gradient assure une excellente séparation, mais a le désavantage d'engendrer un volume d'échantillon résultat très élevé et donc de diluer l'échantillon initial. Par contre, le décrochage par étapes, méthode plus rapide, concentre l'échantillon, mais la purification qui en résulte est moins bonne.

Le système ne dispose d'aucun critère qui lui permette de conseiller une manière plutôt qu'une autre. L'utilisateur est seul juge. Il devra cependant faire connaître son choix au système, car celui-ci aura une influence sur l'efficacité de la purification et sur le volume d'échantillon obtenu à la sortie de la colonne.

6°) Evaluation du volume de l'échantillon recueilli

Le volume probable de l'échantillon après l'application de l'échangeur d'ions dépend du volume du gel et du type de décrochage.

En général, nous pouvons estimer que le volume obtenu sera égal à 50% du volume total du gel dans le cas où l'on effectue le décrochage en une étape. Par contre, si le décrochage est effectué par gradient, le volume probable à la sortie sera égal au volume total du gel.

7°) Evaluation de la nature et de la quantité de protéines dans le résultat

Un échangeur d'ions permettra d'éliminer complètement toutes les protéines dont le PI n'est pas compris dans l'intervalle [PI de la protéine à purifier - fourchette , PI de la protéine à purifier + fourchette]. Les protéines dont le PI est compris dans cet intervalle contamineront d'autant plus le résultat que leur PI est proche du PI de la protéine à purifier.

'Fourchette' est une variable désignant le coefficient de résolution du gel et pouvant prendre un éventail de valeurs (0.1, 0.2, 0.3, 0.5,...).

Afin d'estimer le pourcentage de contamination d'une protéine dont le PI est compris dans l'intervalle, nous considérons que si le PI de la protéine est :

- égal à la limite inférieure ou supérieure de l'intervalle, la contamination est nulle (0%)
- égal au PI de la protéine à purifier, la contamination est maximale (100%)

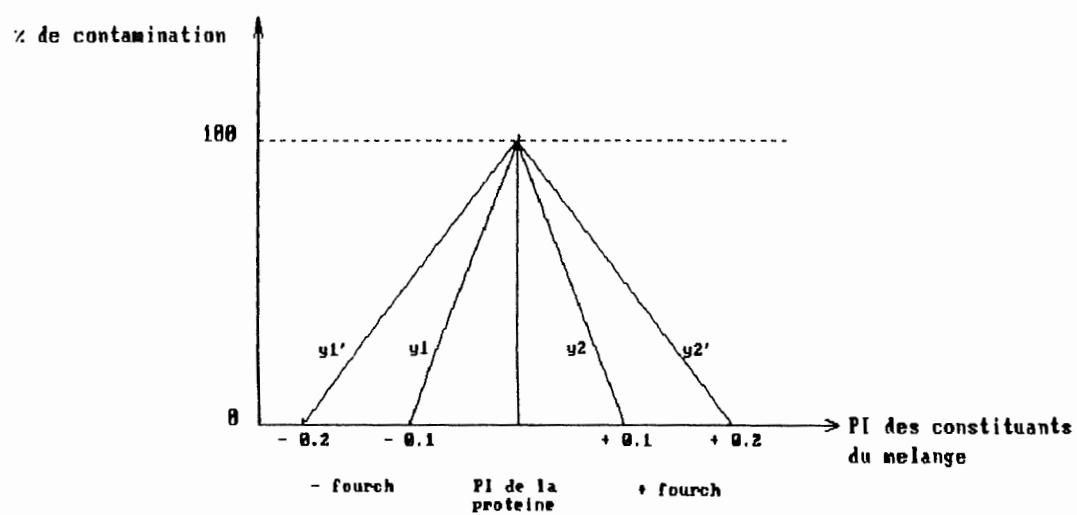


Figure 32 :
 Calcul du pourcentage de contamination des pI dans la
 chromatographie sur échangeur d'ions.
 [source personnelle]

- compris entre la limite inférieure de l'intervalle et le PI de la protéine à purifier, le pourcentage de contamination d'une protéine est une fonction linéaire qui passe par les points de coordonnées (limite inférieure , 0%) et (PI de la protéine à purifier , 100%). Cette équation correspond à une droite dans le plan Y1X :

$$Y1 = ((100 / (PI-(PI-fourchette)))X - (100*(PI-fourchette)) / (PI-(PI-fourchette)))$$

- compris entre le PI de la protéine à purifier et la limite supérieure de l'intervalle, le pourcentage de contamination d'une protéine est une fonction linéaire passant par les points de coordonnées (PI de la protéine à purifier , 100%) et (limite supérieure , 0%). Il s'agit également d'une équation correspondant à une droite dans le plan Y2X :

$$Y2 = ((100 / (PI-(PI+fourchette)))X - (100*(PI+fourchette)) / (PI-(PI+fourchette)))$$

La figure 32 sert de support aux explications données ci-dessus.

8°) Evaluation de la durée de l'échangeur d'ions

La durée moyenne d'un échangeur d'ions est approximativement de 4 heures. Celle-ci peut varier en fonction du débit utilisé.

9°) Evaluation de la perte d'activité de la protéine

La perte d'activité due à chaque manipulation est estimée à environ 15% pour un échangeur d'ions. A cette perte d'activité due à la manipulation, il faudra ajouter la perte résultant de la dénaturation de la protéine au cours du temps pour obtenir la perte totale d'activité.

c. La chromatographie sur tamis moléculaire

c.1. Intérêt de la chromatographie sur tamis moléculaire

Ce type de chromatographie, basé sur la différence de volume hydrodynamique des molécules, est utilisé chaque fois que la protéine à séparer comporte une masse moléculaire différente de celle présumée des protéines contaminantes. Du fait de son pouvoir séparateur relativement faible, de sa lenteur et d'un aspect peu productif, cette technique est en principe appliquée dans les étapes finales de purification où la protéine à séparer est accompagnée d'un nombre limité de contaminants faiblement concentrés.

La filtration sur gel est irremplaçable dans le cas où l'on désire réellement un produit très pur ou lorsque le schéma de purification ne comporte pas de chromatographie d'affinité due à la difficulté d'obtenir un ligand biospécifique.

c.2. Supports d'une chromatographie sur tamis moléculaire

La réalisation d'un tamis nécessite l'emploi d'une colonne, d'un gel et d'un tampon.

Le système doit guider l'opérateur de laboratoire en lui conseillant le(s) support(s) le(s) plus adapté(s) aux conditions de purification.

1°) La colonne

La colonne possède un certain nombre de caractéristiques, à savoir un numéro, un diamètre, une hauteur, une vitesse d'écoulement et un volume maximum d'échantillon pouvant y être déposé. Chaque colonne, de par ses caractéristiques, est plus adaptée à certaines purifications qu'à d'autres.

Détaillons quelque peu certains de ces paramètres. La vitesse d'écoulement est mesurée par le débit spontané de la colonne lorsqu'on réalise une pression en eau sur celle-ci, correspondant au 1/4 de sa hauteur. On utilise une vitesse d'élution correspondant au 3/4 du débit spontané. Chaque colonne accepte un volume d'échantillon maximum ; cette donnée est cruciale car, pour obtenir une bonne séparation, l'opérateur dépose un volume inférieur au volume d'échantillon maximum de la colonne. En tout cas, il ne peut déposer que des échantillons dont le volume est à peine supérieur au volume d'échantillon maximum.

Quatre colonnes sont actuellement disponibles. Le tableau 9 ci-dessous présente leurs caractéristiques.

numéro	diamètre (cm)	hauteur (cm)	vitesse (ml/h)	volume maximum (ml)
1	0.5	80	2	0.25
2	1.0	80	8	1.00
3	2.0	80	15	3.50
4	4.0	80	40	20.00

2°) Le gel

Différents types de gels sont utilisés au laboratoire. Ils se distinguent les uns des autres par leur taux de réticulation et, par conséquent, par leur pouvoir de gonflement. De plus, chaque gel est disponible en grains de différentes grosseurs. Les grains fins sont conseillés pour les travaux de laboratoire et de préparation exigeant une assez bonne résolution. Les gros grains sont eux adaptés aux travaux en cuve. Nous considérerons que les gels connus sont de taille fine.

Chaque gel est caractérisé par son type, sa marque, son domaine de fractionnement et son coefficient de non-séparation.

Le domaine de fractionnement du gel est une zone de poids moléculaire dans laquelle le gel assure une séparation des protéines de poids moléculaire différent. Les molécules dont le poids moléculaire est plus élevé que la limite supérieure de ce domaine - limite d'exclusion - sont totalement exclues du gel. Les molécules dont le poids moléculaire est plus petit que la limite inférieure seront généralement éluées par un volume sensiblement égal au volume du gel (ou volume total).

Un coefficient de non-séparation exprime une estimation de l'efficacité de la séparation réalisée au moyen d'un gel particulier.

Six gels sont connus du système. Leurs caractéristiques respectives apparaissent ci-dessous dans le tableau 10 :

type	marque	domaine de fractionnement (daltons)	coefficient de non-séparation
Sephadex G75	Pharmacia	6000 - 65000	1.5
Sephadex G100	Pharmacia	1000 - 100000	1.5
Sepharose 6B	Pharmacia	10000 - 3000000	3.0
Sephacryl S200	Pharmacia	5000 - 150000	2.0
Ultrogel AcA44	LKB	10000 - 140000	1.5
Ultrogel AcA22	LKB	100000 - 1200000	?

3°) Le tampon

Le tampon généralement utilisé est un tampon phosphate 0.01M à 4°C et de pH 7.4, contenant du NaCl 0.15M. Parfois, on utilise un tampon TRIS-HCl 0.05M de pH 7.4. Le pK permet de caractériser le pouvoir tampon :

- nom : tampon phosphate
pK : 7.2
- nom : tampon TRIS
pK : 8.2.

c.3. Critères pour le choix de l'application de la chromatographie sur tamis moléculaire

Lorsque le système réalise un choix quant à la technique la plus adéquate à appliquer au mélange à purifier, il peut envisager le tamis moléculaire.

Deux critères lui permettent d'estimer l'applicabilité de cette technique :

1- le nombre de substances doit être peu important.

En effet, lorsque le nombre de substances contenues est trop important, la viscosité du mélange augmente et la séparation qui en résulte est très mauvaise. Nous avons opté pour un nombre de substances inférieur ou égal à 15 pour décider de l'applicabilité du tamis.

2- le volume de l'échantillon doit être peu important.

Nous avons vu précédemment que le volume d'un mélange à déposer sur une colonne de tamis moléculaire est limité. Nous pouvons énoncer des règles fort générales à propos du volume :

- plus le volume du mélange est petit, meilleures sont les conditions pour l'application d'un tamis.
- la technique, telle qu'elle est utilisée au laboratoire, accepte des mélanges d'un volume maximum avoisinant 18 ml.

Le système doit s'assurer que le volume du mélange respecte la règle énoncée ci-dessus : le volume doit être le plus petit possible, de préférence inférieur à 18 ml.

Ceci peut amener le système à suggérer une concentration de l'échantillon sur une membrane d'ultrafiltration de façon à ramener le volume dans des limites acceptables. Il est établi que la concentration idéale de matériel dans l'échantillon est de 10 mg/ml. Dès lors, nous avons adopté les règles suivantes :

- si la concentration du mélange est inférieure à 10 mg/ml (concentration idéale), le système conseille l'application d'un tamis moléculaire ou de concentrer l'échantillon afin de tendre vers la concentration idéale,

- si la concentration du mélange est égale à la concentration idéale, le système opte pour le tamis moléculaire,
- si la concentration du mélange est supérieure à la concentration idéale alors, le programme proposera d'appliquer un tamis moléculaire ou de diluer l'échantillon.

c.4. Evaluation de l'application de la chromatographie sur tamis moléculaire

1°) Choix du tampon

Le domaine de "pouvoir tampon" s'étend sur l'intervalle $[pK-1, pK+1]$. Le système doit conseiller l'utilisation d'un tampon dont le domaine de "pouvoir tampon" est compatible avec le domaine de stabilité de la protéine à purifier. Il faut travailler à un pH de 7.4, si possible, assurant la stabilité de la protéine à purifier.

Le choix entre les deux tampons disponibles se réalise de la façon suivante :

- si le P.I. est inférieur ou égal à 7.2, alors le tampon conseillé est le phosphate,
- si le P.I. appartient à l'intervalle $[7.2, 8.2]$, alors les tampons possibles sont le phosphate ou le TRIS,
- si le P.I. est supérieur à 8.2, alors le tampon le plus adéquat est le TRIS.

2°) Choix du gel

Le choix du gel s'effectue selon une procédure complexe que nous allons tenter de vous expliquer. Pour les différents gels utilisés au laboratoire, nous connaissons, outre leur domaine de fractionnement (c'est-à-dire les limites inférieure

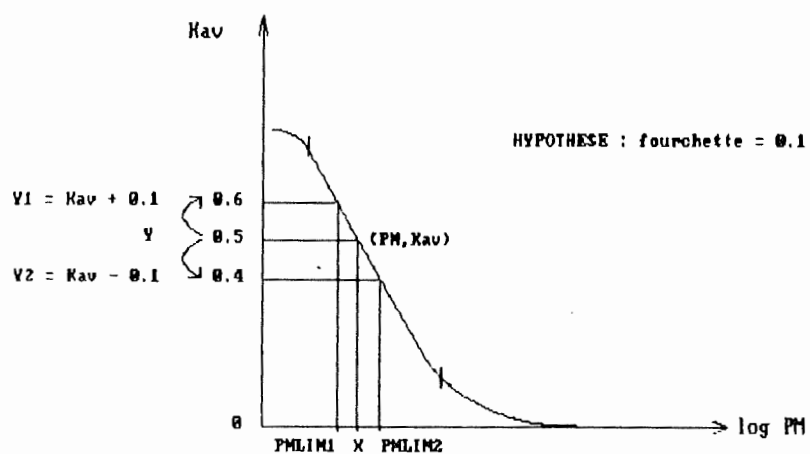


Figure 33 :
Evolution du $\log(PM)$ en fonction du Kav (tamis moléculaire et HPLC).
[source personnelle]

et supérieure d'exclusion), leur droite d'étalonnage (relation existant entre Kav et log PM).

La figure 4 présentait l'évolution du Kav en fonction du log PM pour les Sephadex G75 et G100.

Afin de cerner l'éventail des gels pouvant être utilisés, nous demandons à l'utilisateur d'entrer une "fourchette" de Kav s'étendant de 0.1 unité à 0.5 unité. Notons cependant que 0.1 unité est la valeur la plus fréquemment utilisée.

Notons qu'ici aussi, pour rester cohérentes avec le programme, l'emploi de notations indicées (Kav) ne c'est pas avéré nécessaire.

Voici le procédé de travail :

- Pour chaque gel, nous avons calculé l'équation de la partie linéaire de la courbe d'étalonnage.
- Etant donné le PM de la protéine à purifier, le programme calcule par le biais de cette équation le Kav de la protéine (appelons-le Y).
- A cette valeur Y, on ajoute dans un premier temps la valeur de la "fourchette" : supposons celle-ci égale à 0.1 pour des raisons de facilité, et l'on obtient ainsi Y1 (exemple : $Y + (0.1) = Y1$); puis, dans un second temps on retranche de Y cette même valeur de "fourchette", et l'on obtient alors Y2 (exemple : $Y - (0.1) = Y2$).
- Y1 et Y2 nous permettent donc d'obtenir deux nouvelles valeurs sur l'axe des X, et qui sont respectivement PMLIMX1 et PMLIMX2. Ces dernières constituent les poids moléculaires limites calculés (voir figure 33).

Illustrons par un exemple concret le raisonnement qui précède ; et ce pour deux des gels couramment utilisés au laboratoire, à savoir le Sephadex G75 et le Sephadex G100.

Les équations des parties linéaires des courbes d'étalonnage du Sephadex G75 et du Sephadex G100 sont les suivantes :

Pour le Sephadex G75 : $y = 2.8674 - 0.2549 * \log (PM)$

Pour le Sephadex G100 : $y = 3.0735 - 0.2609 * \log (PM)$

Supposons que la protéine à purifier possède un poids moléculaire de 45000 daltons et une "fourchette" de K_{av} égale à 0.1.

Avec le Sephadex G75 : $Y = K_{av} = 0.136$
 $Y1 = (Y + 0.1) = 0.236$
 $Y2 = (Y - 0.1) = 0.036$
 $PMLIMX1 = 30432 \text{ daltons}$
 $PMLIMX2 = 66695 \text{ daltons.}$

Avec le Sephadex G100 : $Y = K_{av} = 0.278$
 $Y1 = (Y + 0.1) = 0.378$
 $Y2 = (Y - 0.1) = 0.178$
 $PMLIMX1 = 30685 \text{ daltons}$
 $PMLIMX2 = 66047 \text{ daltons.}$

Dès lors, nous pouvons dire :

Avec le Sephadex G75 : toutes les protéines dont le PM est inférieur à 30432 daltons ou supérieur à 66695 daltons seront éliminées ; les protéines dont le PM est compris entre ces deux bornes contamineront la protéine à purifier.

Avec le Sephadex G100 : toutes les protéines dont le PM est inférieur à 30685 daltons ou supérieur à 66047 daltons seront éliminées ; les protéines dont le PM est compris entre ces deux bornes contamineront la protéine à purifier.

Notons que ces calculs (Y, Y1, Y2, PMLIMX1, PMLIMX2) n'ont pas toujours de sens. En effet, si Y1 est négatif (et à fortiori Y2 aussi), cela signifie que le PM de la protéine se situe en dehors des limites de séparation du gel ; ce dernier n'est donc d'aucune utilité, et par conséquent, calculer PMLIMX1 et PMLIMX2 serait vain et même erroné. De plus, si Y2 est négatif (Y1 pouvant être positif), cela signifie que le gel ne séparera que les contaminants dont le PM est inférieur au PM de la protéine à purifier.

Il reste cependant un problème. Quand plusieurs gels semblent convenir (cfr. exemple ci-dessus), lequel le système va-t-il conseiller ?

Deux critères sont ici appliqués :

- les PM limites calculés (PMLIMX1 , PMLIMX2) doivent appartenir au domaine de fractionnement du gel.
- la différence entre les PM limites calculés doit être la plus faible possible.

Dès lors, dans l'exemple exposé ci-dessus, nous obtenons :

Pour le Sephadex G75 : $66695 - 30432 = 36263$ daltons.

Pour le Sephadex G100 : $66047 - 30685 = 35362$ daltons.

Par conséquent, le système conseillera de préférence à l'utilisateur le Sephadex G100.

Il faut cependant noter que l'utilisateur peut ne pas être satisfait par cette proposition. Il a ensuite la possibilité de choisir un autre gel disponible et connu du système, ou d'introduire un nouveau gel. Pour ce faire, il doit en donner une description complète (type, marque, domaine de fractionnement et coefficient de non-séparation).

3°) Choix de la colonne

Le volume du mélange à déposer sur la colonne est connu. Le système réalise le choix de la colonne et ce, sur base de l'unique critère suivant : il choisira la plus petite colonne acceptant un échantillon d'un tel volume.

Ce choix réalisé parmi l'ensemble des colonnes connues du système est soumis à l'utilisateur qui peut l'accepter ou le refuser. Dans ce cas, la liste des colonnes disponibles (ainsi que leurs caractéristiques) s'affiche et l'utilisateur peut choisir une autre colonne, ou en introduire une nouvelle. Dans cette éventualité, l'utilisateur doit entrer les caractéristiques complètes de cette nouvelle colonne afin qu'elle puisse être mémorisée par le système.

4°) Evaluation du volume de l'échantillon recueilli

L'utilisateur ayant dû dans un premier temps entrer le volume de l'échantillon, ainsi que le volume du gel, le système peut maintenant évaluer le volume de l'échantillon résultat. Cette évaluation est basée sur la formule suivante :

$$\text{volume résultat} = (5 \% \text{ volume du gel}) + \text{volume échantillon.}$$

5°) Evaluation de la nature et de la quantité des substances dans l'échantillon recueilli

Lorsque le système a décidé l'application du tamis moléculaire, il a fait le choix du gel et de la colonne. Dès lors, les poids moléculaires limites calculés sont connus. Ces derniers vont nous permettre d'estimer le pourcentage de contamination des protéines contaminantes. Le tamis moléculaire permettra d'éliminer complètement toutes les substances dont le PM est soit supérieur à PMLIMX2, soit inférieur à PMLIMX1. Les protéines dont le PM est compris dans le domaine [PMLIMX1 ,

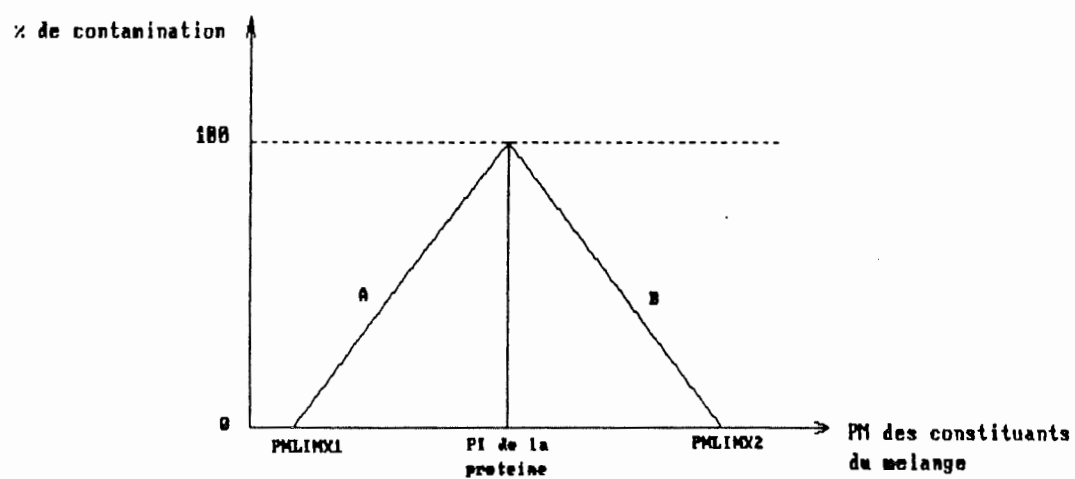


Figure 34 :
 Calcul du pourcentage de contamination des PM dans le tamis
 moléculaire et dans l'HPLC.
 [source personnelle]

PMLIMX2] contamineront d'autant plus le résultat que le poids moléculaire est proche du PM de la protéine à purifier.

Afin d'estimer le pourcentage de contamination d'une substance dont le poids moléculaire est compris dans l'intervalle [PMLIMX1 , PMLIMX2], nous considérons que si le poids moléculaire de la substance est :

- égal à la limite inférieure ou supérieure de l'intervalle, la contamination est inexistante (0%),
- égal au poids moléculaire de la protéine à purifier, la contamination est maximale (100%),
- compris entre la limite inférieure de l'intervalle et le poids moléculaire de la protéine à purifier, le pourcentage de contamination d'une substance est une fonction linéaire passant par les points de coordonnées (limite inférieure , 0%) et (PM de la substance à purifier , 100%).

L'équation de cette droite sera par conséquent :

$$A = \frac{100}{(PM-PMLIMX1)} X - \frac{100 \cdot PMLIMX1}{(PM-PMLIMX1)}$$

- compris entre le poids moléculaire de la protéine à purifier et la limite supérieure du domaine, le pourcentage de contamination d'une substance est une fonction linéaire passant par les points de coordonnées (PM de la protéine à purifier , 100%) et (limite supérieure , 0%).

Ici aussi une équation de droite peut se calculer :

$$B = \frac{100}{(PM-PMLIMX2)} X - \frac{100 \cdot PMLIMX2}{(PM-PMLIMX2)}$$

La figure 34 illustre le raisonnement suivi. Après que ce calcul ait été réalisé pour chacune des protéines contaminantes du mélange, l'utilisateur entre une proportion qui

indique la quantité de protéine contenue dans le mélange ; ceci permet donc au système de calculer un pourcentage global de contamination pour chacun des contaminants du mélange.

6°) Evaluation de la durée du tamis moléculaire

La durée du tamis moléculaire est fonction de la colonne utilisée pour le réaliser. Cette durée varie entre 15 et 20 heures en général.

7°) Evaluation de la perte d'activité de la protéine

La perte d'activité due à la manipulation est estimée à 10% pour un tamis moléculaire. A cette perte d'activité, il faudra ajouter la perte d'activité provenant de la dénaturation de la protéine au cours du temps pour obtenir la perte totale d'activité.

d. L'HPLC

Cette technique de purification, qui n'est autre qu'un tamis moléculaire à "grande vitesse" (analytique), permet au début d'un processus de purification de déterminer les poids moléculaires approximatifs des contaminants de la protéine à purifier.

Le seul critère applicable pour guider l'utilisateur vers l'application d'une HPLC est le cas où celui-ci ignore les poids moléculaires des autres constituants du mélange.

Les choix du gel et du tampon s'effectuent de manière similaire (c'est-à-dire se basent sur les mêmes critères) au tamis moléculaire (cfr. c.).

Toutefois, l'utilisateur doit entrer le "nom" de la colonne (exemples : Superose, TSK) ainsi que son débit (de l'ordre de 0.4-0.5 ml/min), car ce dernier paramètre permettra de calculer ultérieurement la durée de cette technique. Il est également demandé à l'utilisateur d'introduire le volume du gel (environ une dizaine de ml) et le volume de l'échantillon qu'il compte déposer au sommet de la colonne. Il est à noter que ce volume ne peut excéder 0.1 ml.

Le calcul du volume résultat de l'HPLC est basé sur la même formule que celle employée pour le tamis moléculaire.

En ce qui concerne la durée de cette technique, nous évaluons cette dernière de la façon suivante :

$$\text{durée HPLC} = (\text{volume du gel} + \text{volume échantillon}) / \text{débit HPLC}$$

Le pourcentage de contamination des protéines contaminantes se calcule également de manière identique au tamis moléculaire. La perte d'activité due à la réalisation d'une HPLC est voisine de 10%. Notons qu'à cette perte d'activité inhérente à la technique, il faut ajouter la perte résultant de la dénaturation de la protéine au cours du temps.

e. L'électrofocalisation

Cette technique de séparation des protéines possède deux rôles fondamentaux dans une purification.

Au début de cette dernière, une électrofocalisation s'impose afin de déterminer les points isoélectriques des contaminants de la protéine à purifier. Il s'agit donc ici d'un usage purement analytique. Suite à la réalisation de cette technique, il est demandé à l'utilisateur d'entrer les P.I. des contaminants de la protéine à purifier.

Un autre usage, préparatif celui-là, existe pour l'électrofocalisation. Le critère permettant d'appliquer cette technique est que les P.I. des protéines contaminantes doivent être différents de 0.1 unité au moins. Par conséquent, cette longue technique (plus d'un jour) ne se réalisera que dans la mesure où l'opérateur désire un haut degré de pureté de la protéine, ou qu'il reste dans le mélange une protéine 'très' contaminante et donc difficile à éliminer par les techniques chromatographiques. Typiquement, l'électrofocalisation préparative se réalisera donc à la fin d'un processus de purification.

f. L'électrophorèse

L'électrophorèse (basée sur un principe de séparation identique à celui de l'échangeur d'ions) ne possède que peu de critères d'application bien définis. Le seul que nous puissions raisonnablement émettre est que pour réaliser une électrophorèse, il faut une différence de 0.5 unité entre les PI des protéines contaminantes.

Nous observons donc que l'électrophorèse effectue une séparation moins "fine" des protéines que l'électrofocalisation, technique qui lui est similaire par de nombreux aspects.

Après la réalisation de cette technique, l'utilisateur introduit les différents points isoélectriques obtenus par l'application de la technique. Par la suite, le système calcule, pour chaque contaminant, son pourcentage de contamination et ce, grâce à la même procédure que celle employée pour la chromatographie sur échangeur d'ions (cfr. b.).

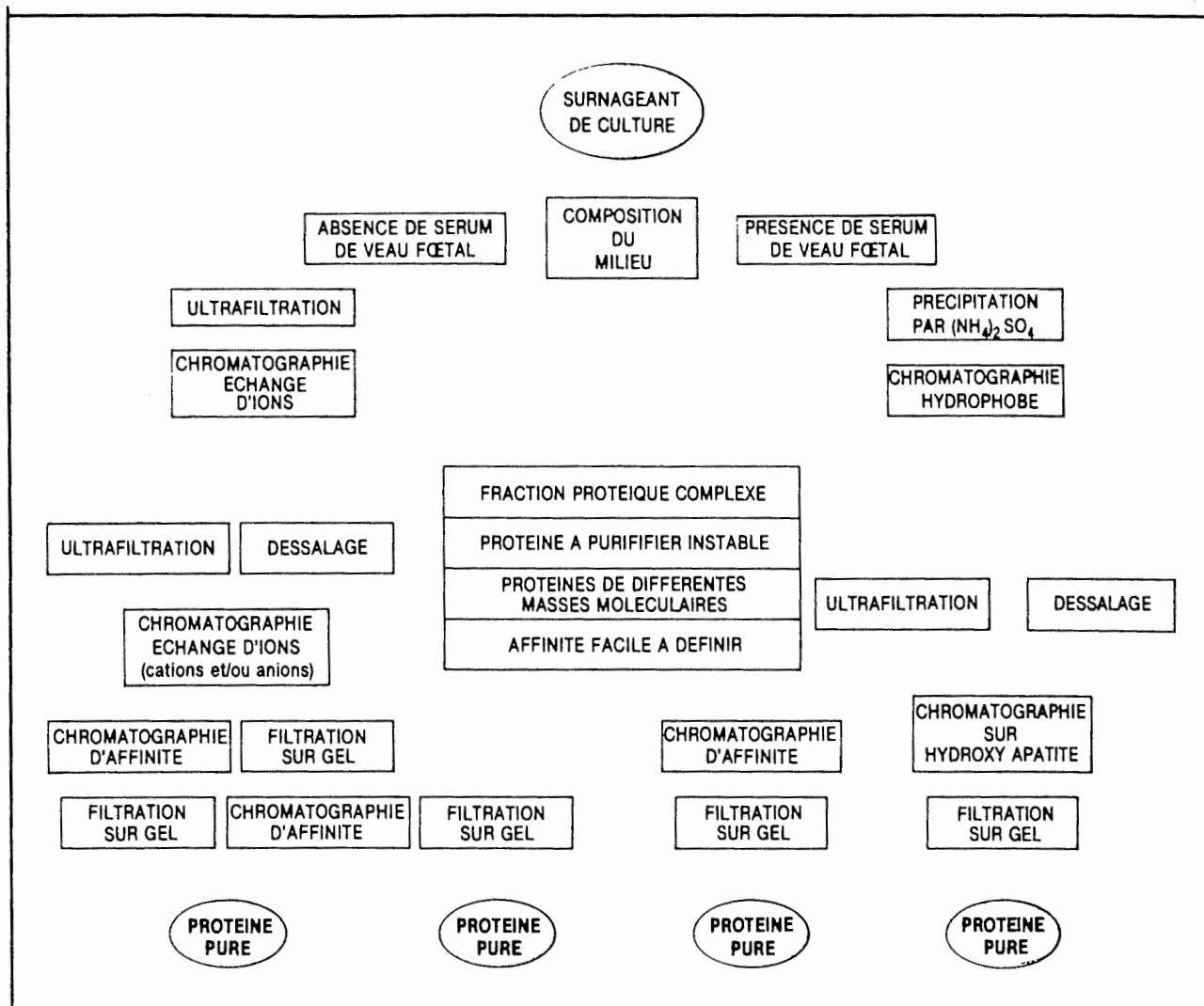


Figure 35 :
Schéma de purification d'une protéine sécrétée par des cellules en culture en fonction du contexte.
[Boschetti E., 1987]

2.2.4. Les enchaînements des techniques de purification

Nous venons de voir les critères permettant d'orienter le processus de purification vers telle ou telle autre technique de purification.

Cependant, la mise en oeuvre opérationnelle de ce processus nous amène à considérer que chaque étape n'est pas totalement isolée : elle est fortement influencée par l'étape qui précède et influence de manière décisive l'étape suivante. D'autre part, la bonne cohérence entre les différentes étapes dépend souvent de la nature du milieu de culture. Celui-ci doit être défini en fonction de la protéine attendue et de l'efficacité que l'on peut attendre des techniques de purification mises en oeuvre afin d'isoler la protéine en question.

a. Un exemple d'enchaînement logique des méthodes de séparation

E.Boschetti [BIO87] tente de donner quelques règles générales (voir figure 35) concernant l'enchaînement des techniques de purification.

Avant toute opération d'extraction ou de purification, il est nécessaire de séparer les cellules vivantes du surnageant. Cette opération est aisément effectuée par simple centrifugation ou filtration. Ensuite, une précipitation au sulfate d'ammonium peut être réalisée. Le précipité redissout dans un volume minimum de tampon se trouve dans un environnement fortement salin tout à fait adapté à la mise en oeuvre d'une chromatographie hydrophobe. La protéine à séparer est, par cette étape, débarrassée d'une partie des protéines contaminantes et son isolement peut maintenant être poursuivi par d'autres étapes chromatographiques.

Cette ligne de conduite (précipitation puis chromatographie hydrophobe) peut ne pas être la plus appropriée. Lorsque le surnageant de culture est pauvre en protéines, il est souhaitable de procéder à une concentration par ultrafiltration, laquelle permettra d'éliminer les petites molécules et, de surplus, d'équilibrer les protéines dans le tampon choisi pour l'étape chromatographique suivante. A ce stade, la chromatographie la plus appropriée sera l'échangeur d'ions et principalement l'échangeur d'anions (DEAE) en raison de sa bonne sélectivité, de sa grande capacité et de sa facilité de mise en oeuvre.

A présent, le choix en ce qui concerne les techniques qui vont s'enchaîner se fera en fonction de plusieurs critères :

- complexité de la fraction à purifier (nombre de protéines),
- diversité des masses moléculaires,
- stabilité de la protéine à purifier,
- disponibilité d'une méthode d'affinité (ligand spécifique).

Une fraction protéique encore trop complexe pour être soumise à une filtration sur gel sera soumise de préférence à une chromatographie d'échange d'ions. Malheureusement, cette approche nécessite de nouveau un rééquilibrage de la fraction avec un tampon compatible. Les solutions possibles pour résoudre ce problème sont la diafiltration, la dialyse ou le dessalage. A partir de ce stade, une filtration sur gel et/ou une chromatographie d'affinité suivront dans un ordre encore à définir.

Une fraction constituée d'un nombre très limité de protéines peut être soumise à une filtration sur gel. Aucun traitement préalable (dessalage) de la fraction à purifier n'est nécessaire.

Dans cet ensemble de techniques, il faut tout particulièrement veiller à la compatibilité des enchainements chromatographiques.

b. Critères d'enchainement des techniques de purification

Comme il a déjà été dit, le système débute fréquemment par l'application d'une Hplc et/ou d'une électrofocalisation, et ce étant donné que la plupart du temps les P.M. et P.I. des contaminants de la protéine à purifier sont inconnus.

De ces techniques de séparation "grossière", il résulte deux informations qui sont le volume de l'échantillon et le pourcentage total de dénaturation de la protéine. Dès lors, ces résultats permettront l'enchainement avec d'autres techniques. Le volume résultat de l'Hplc deviendra le volume échantillon de la technique suivante. Le pourcentage total de dénaturation de la protéine devra être "cumulé" suite aux différentes techniques réalisées.

Malheureusement, il est souvent impossible de faire suivre une chromatographie par une autre de manière directe pour deux raisons principales :

- la nature du tampon dans lequel se trouve l'échantillon,
- son degré de dilution.

Le passage d'une technique chromatographique à une autre se fait parfois au prix d'une adaptation de l'échantillon (dessalage,...) et éventuellement d'une concentration préalable.

Les principales erreurs à ne pas commettre sont :

- appliquer un échantillon dilué sur une colonne de filtration sur gel,
- utiliser un tamis moléculaire à un stade précoce du schéma de purification.

c. Perspectives de développement

Nous avons vu succinctement comment peut être défini un schéma de purification en associant la chromatographie sous ses différents aspects à d'autres techniques séparatives telle que la précipitation.

En fonction de la nature de l'échantillon, de nombreux schémas peuvent être imaginés, les règles générales d'application étant grossièrement les mêmes. La purification d'une protéine, quelle que soit son origine, n'est pas toujours facile à effectuer, tant le choix des techniques est vaste et complexe. D'autre part, des critères rationnels et précis seraient souhaitables afin de standardiser les procédures de laboratoire et faciliter leur traitement par l'ordinateur.

2.3. Implémentation

2.3.1. Moteur d'inférence

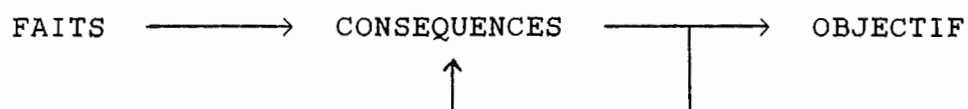
Comme il l'a été développé au chapitre 2, la connaissance stockée dans un système expert est habituellement représentée sous forme d'un ensemble de règles appelé base de connaissances. Celle-ci représente une perception de la connaissance sur un domaine donné qui est beaucoup plus flexible que celle d'un programme conventionnel.

Cette connaissance est manipulée par un moteur d'inférence qui représente, quant à lui, la manière de penser de l'homme.

On peut également dire que le moteur d'inférence est un interpréteur de règles.

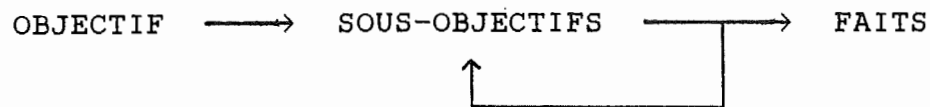
Il peut les prendre en considération de trois façons différentes:

- en partant des faits: il passe donc des faits à des conséquences de faits; ensuite de ces conséquences à des conséquences de conséquences, ... jusqu'à obtenir l'objectif, si c'est possible.
Si les prémisses sont vraies, la règle est tirée; dans le cas contraire, elle est abandonnée.
On parle de chainage avant (ex: OPS5).



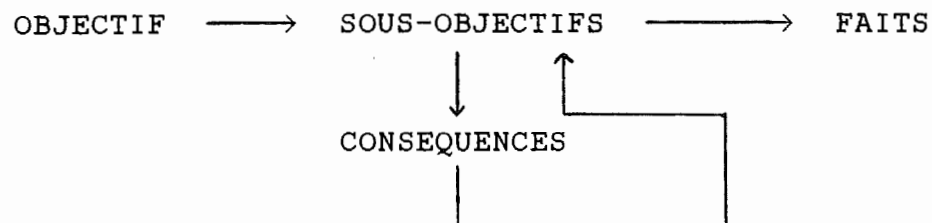
- en partant de l'objectif: le moteur passe de l'objectif à des sous-objectifs, ... jusqu'à aboutir aux faits si c'est possible.

On parle de chaînage arrière (ex: Prolog, VP-Expert).



La plupart des petits et moyens shells utilisent ce type de chaînage pour résoudre des problèmes de diagnostic ou de monitoring.

- en partant soit de l'objectif, soit des faits; on parle de chaînage mixte (ex: Nexpert, Guru).



Ce type de chaînage permet d'utiliser la base de connaissance d'une façon plus optimale que les deux autres méthodes; le système est effectivement capable de trouver la plupart des valeurs à assigner à des variables, alors qu'en chaînage avant ou arrière, il serait dans l'obligation de les demander à l'utilisateur. Par exemple, si les prémisses d'une règle sont vraies, la conclusion est tirée; sinon, elle est abandonnée. Si une ou plusieurs variables de la condition sont inconnues, le système essaie de leur trouver une valeur par chaînage arrière.

La méthode sur laquelle se base le raisonnement du moteur d'inférence VP-Expert est le chaînage arrière. Le moteur commence donc par identifier la variable but, et parcourt la séquence de règles jusqu'à ce qu'il trouve une valeur susceptible de l'aider à assigner une valeur à la variable but.

Autrement dit:

ACTIONS

FIND A;

la variable but est identifiée: A

RULE 1

IF B = VAL

THEN A = VAL;

la première règle possédant A dans sa conclusion est trouvée, cependant B est inconnu. Le moteur va alors chercher la première règle contenant B dans sa conclusion.

RULE 2

IF C = VAL

THEN B = VAL;

même raisonnement avec la variable C

Ce principe se poursuit jusqu'à ce qu'une des variables rencontrées ait une valeur connue. Une fois cela fait, le mécanisme d'inférence peut "retracer", remonter sur ses pas et tester les règles parcourues antérieurement.

Notons que Prolog et des outils comme Guru recensent d'abord l'ensemble des règles susceptibles d'établir une valeur pour le but, dites règles candidates; le système retient alors une de ces règles selon une stratégie de sélection (cfr point 3.3.2. au chapitre 2).

Si la condition de cette règle est vérifiée, la règle est tirée; sinon le système examine la règle candidate suivante.

2.3.2. Stockage des données

Comme il a déjà été dit antérieurement, VP-Expert offre la possibilité d'interagir avec des fichiers externes. En ce qui nous concerne, nous avons travaillé avec le tableur Lotus 1-2-3 et le gestionnaire de bases de données DBase III.

Dans un premier temps, l'emploi du tableur se révélait intéressant, notamment pour le stockage des paramètres définissant les différentes droites d'étalonnage et pour le calcul des PM limites. Mais il s'est avéré impossible de mettre à jour des données dans le tableur et de le manipuler au sein du programme, et ce lors d'une même consultation. Or, ceci constituait l'attrait fondamental de l'utilisation d'un tableur. Nous avons donc dû trouver un subterfuge et, par conséquent, réaliser ces opérations dans le programme lui-même. En fin de course, il apparaît que nous avons complètement abandonné l'idée du tableur, ce dernier ne répondant pas à nos objectifs.

Dans un second temps, l'usage d'un gestionnaire de bases de données s'est révélé, quant à lui, totalement satisfaisant. Nous y avons constitué quatre fichiers que nous allons brièvement décrire. Notons simplement que l'extension ".dbf" est mise pour "Data Base File".

"FIPROTRI.DBF" reprend les caractéristiques des protéines, à savoir leurs :

- nom,
- nature,
- PM,
- PI,
- bornes inférieure et supérieure de stabilité (bi_stab, bs_stab),
- bornes inférieure et supérieure de précipitation (bi_prec, bs_prec) et
- pourcentage de dénaturation (pour_stab).

Ce fichier regroupe actuellement les renseignements relatifs à 36 protéines, principalement des protéines sanguines. Une option de notre programme prévoit d'en ajouter. Voici la liste (provisoire) de ces protéines :

actine G	fibrinogène	ovalbumine
adh	g6 pdh	phosphatase acide
albumine	haptoglobine	phosphatase alcaline
antitrypsine	hémoglobine A	phospholipase A
arginine décarboxylase	hydrogénase	plasminogène
betaglobuline	igg	préalbumine
carboxypeptidase A	insuline	prothrombine
catalase	luciférase	pyruvate kinase
céruloplasmine	lysine	transferrine
cholinestérase	lysosyme	trypsine
chymotrypsinogène	monoamine oxydase	trypsinogène
dnase 2	nucléase	uréase

"FICHCOLT.DBF" présente les propriétés des colonnes disponibles pour l'application du tamis moléculaire et de l'HPLC. Chaque colonne est décrite en termes de ses

- numéro,
- diamètre,
- hauteur,
- vitesse et
- volume maximum de l'échantillon à déposer (vol_max).

Ici aussi, l'utilisateur a la possibilité d'allonger la liste actuellement stockée.

"FICHGELT.DBF" contient les informations relatives aux gels utilisés lors du tamis moléculaire et de l'HPLC :

- type du gel,
- marque,
- limites inférieure et supérieure du domaine de fractionnement (domfracti, domfracts) et
- coefficient de non-séparation (coeffns).

L'ajout d'un nouveau gel est également prévu par le programme.

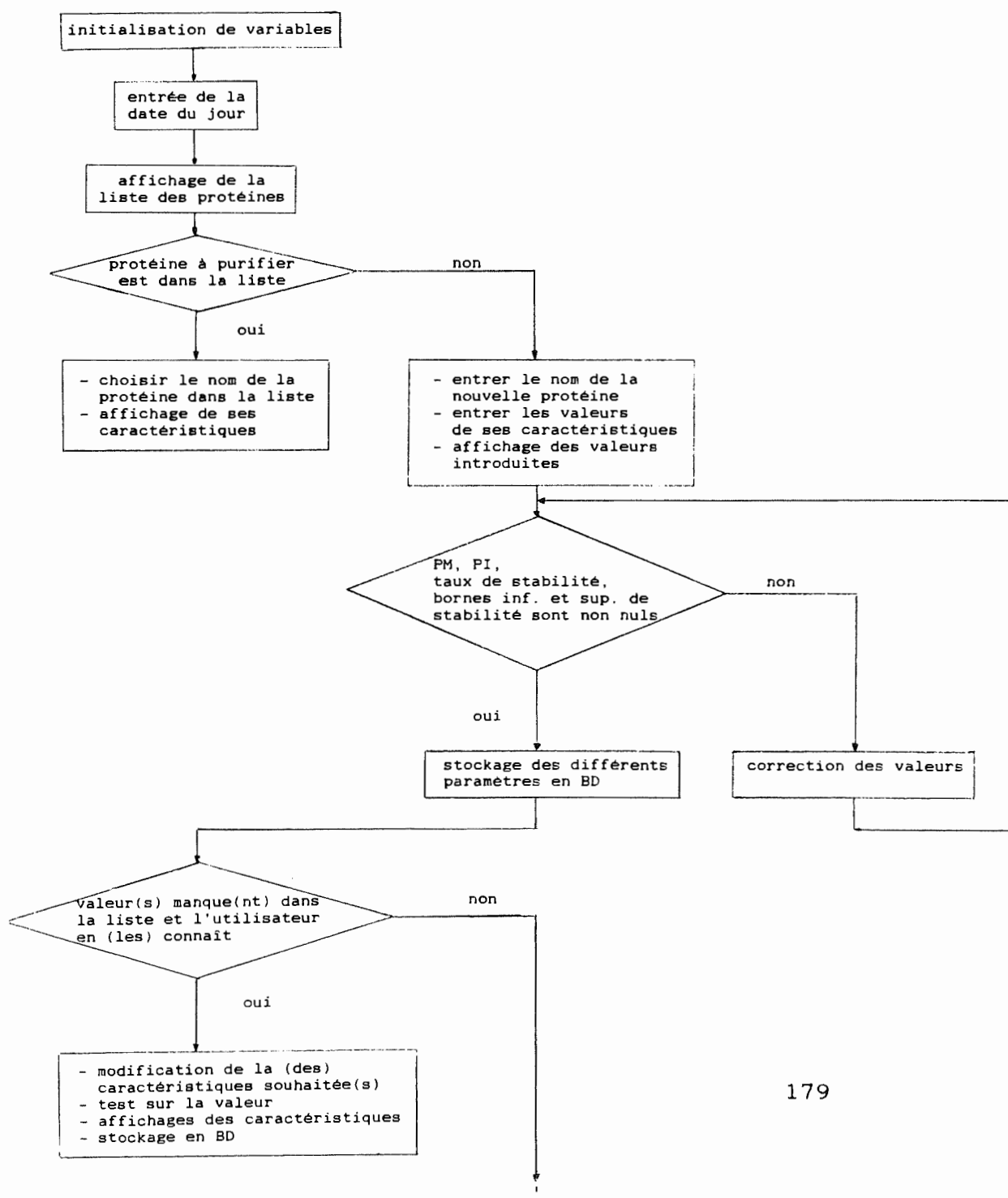
"FICHRES.DBF" est un fichier récapitulatif reprenant divers éléments, soit introduits par l'utilisateur, soit évalués par le logiciel. L'utilisateur a la possibilité de choisir la date du jour pour lequel il souhaite obtenir des renseignements. Ce fichier synthétise pour chaque technique les caractéristiques suivantes :

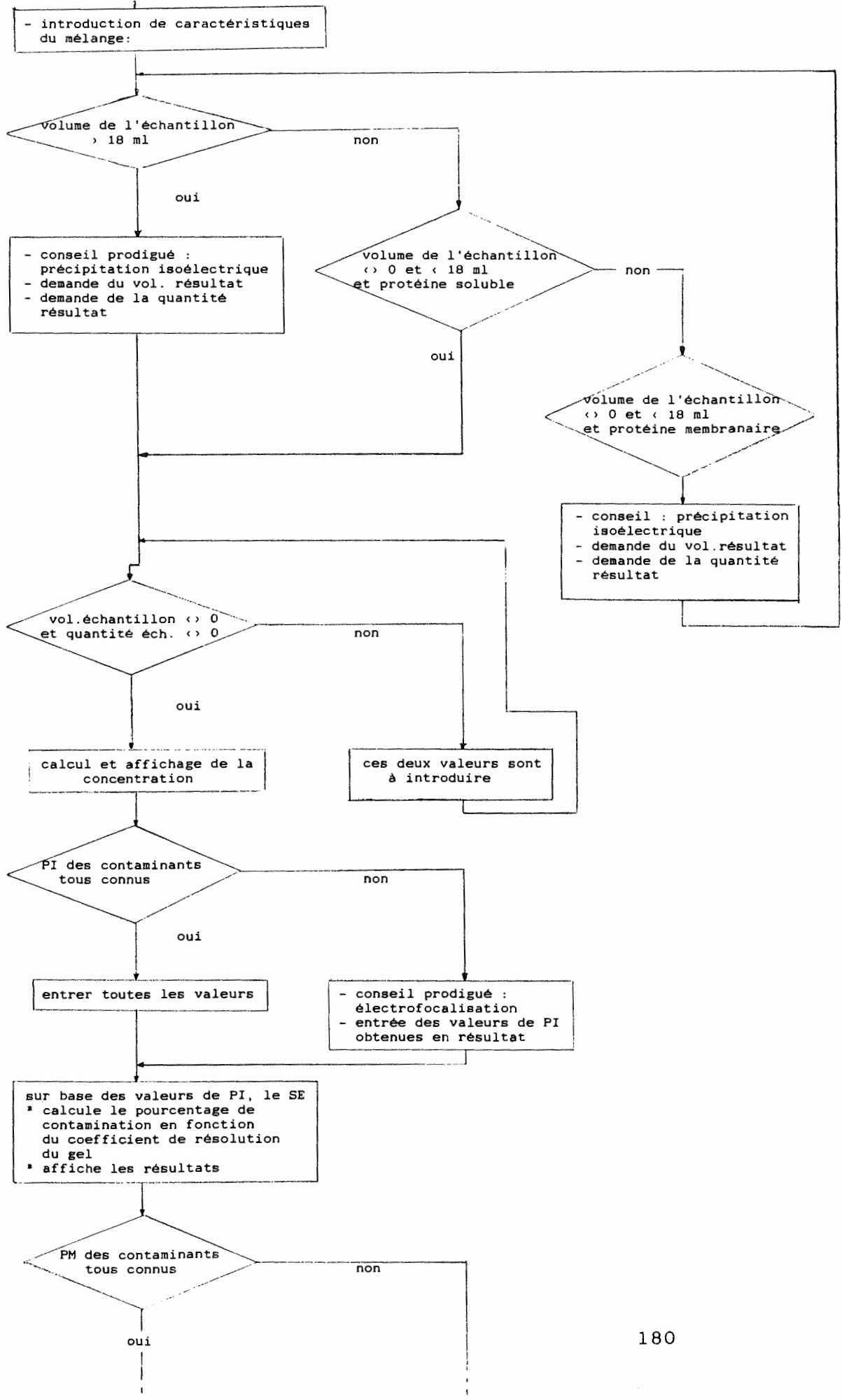
- date du jour demandé (date),
- technique (nomtech),
- gel,
- volume du gel (volgel),
- nom de la colonne (colonne),
- diamètre de la colonne (diamcol),
- tampon,
- pH,
- type d'élution (typelut),
- durée de la technique (durtech),
- perte inhérente à la technique (inhtech),
- volume résultat (volres),
- concentration (concent),
- perte totale due à la technique (pertegl).

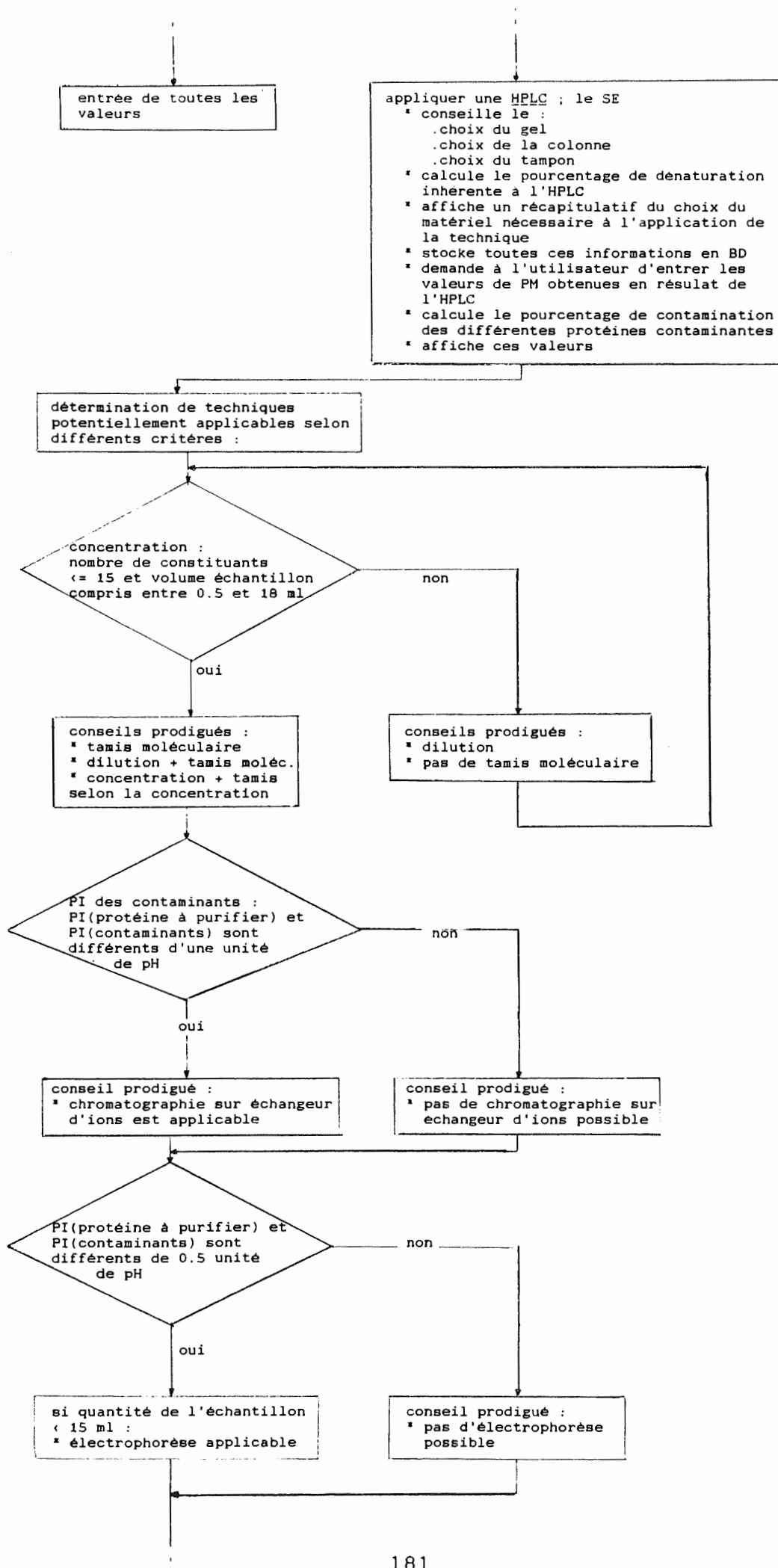
Enfin, un écran de présentation du mémoire a été implémenté en Pascal et sauvé sous une extension ".com". Par cela, nous avons testé la capacité de VP-Expert de faire appel à des fichiers externes via un appel au système.

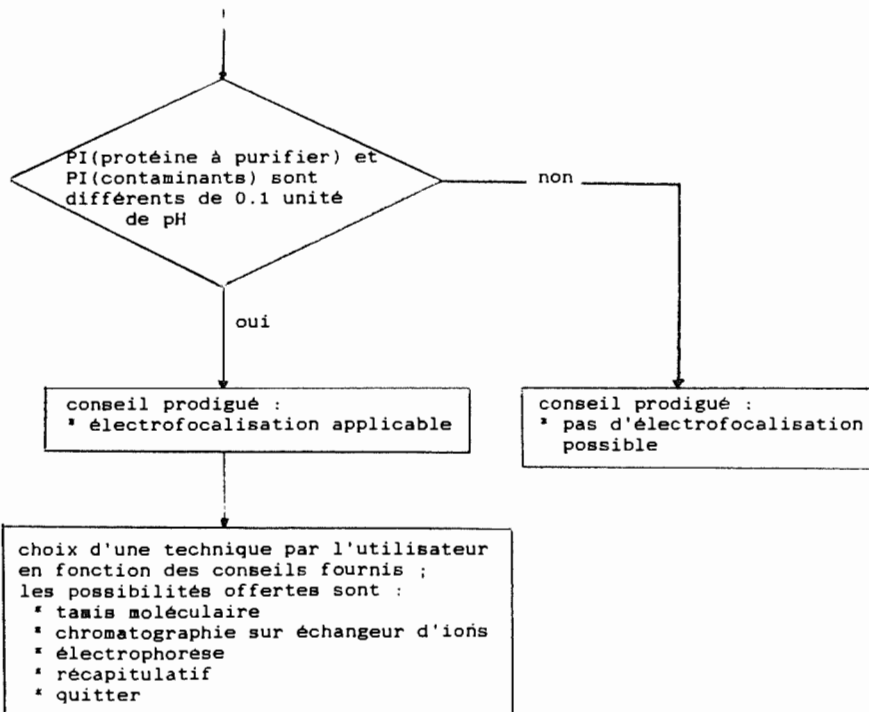
2.3.3. Composition du programme

Dans cette section, nous allons nous attacher à décrire les principales étapes du programme sous forme d'un pseudo-organigramme.









si choix = tamis moléculaire
le SE :

- * conseille un gel
- * conseille une colonne et affiche ses caractéristiques, avec possibilité à l'utilisateur d'effectuer un autre choix
- * demande à l'utilisateur d'introduire des informations (volume du gel, quantité d'échantillon à y déposer)
- * calcule la concentration
- * calcule le volume résultat
- * détermine le tampon à utiliser
- * détermine la durée de la technique
- * calcule le pourcentage de dénaturation inhérente à la technique, ainsi que le pourcentage correspondant à l'ensemble des techniques effectuées jusque là
- * affiche un récapitulatif de ces informations (gel, colonne et ses caractéristiques, tampon, volume résultat, durée, pertes inhérentes au tamis et pertes totales)
- * stocke ces informations en BD
- * demande si une HPLC a été réalisée antérieurement
 - si oui, le SE demande si l'utilisateur désire conserver les valeurs de PM introduites alors
 - si oui, le SE poursuit son investigation
 - sinon * il demande d'introduire les valeurs de PM obtenues en résultat du tamis moléculaire
 - * il calcule le degré de contamination des protéines contaminantes
 - sinon * il demande d'introduire les valeurs de PM obtenues en résultat du tamis moléculaire
 - * il calcule le degré de contamination des protéines contaminantes
- * propose les techniques applicables après un tamis moléculaire en fonction des caractéristiques de l'échantillon (volume résultat, concentration, tampon)
 - si volume résultat < 20 ml
 - alors un tamis est à nouveau applicable
 - sinon il faut concentrer avant de poursuivre
 - si le PI de la protéine à purifier et le PI de tous les contaminants sont différents d'une unité de pH
 - alors une chromatographie sur échangeur d'ions est applicable
 - sinon elle n'est pas applicable
- * affiche le menu des choix de techniques

si choix = chromatographie sur échangeur d'ions
le SE :

- * détermine le pH de travail
- * conseille un gel en fonction du pH
- * détermine le volume du gel
- * conseille un tampon en fonction du pH de travail
- * détermine le diamètre de la colonne
- * détermine l'etyp d'élution de la colonne
- * calcule le volume résultat
- * calcule le pourcentage de dénaturation inhérente à la technique, ainsi que le pourcentage correspondant à l'ensemble des techniques effectuées jusque là
- * affiche un récapitulatif de ces informations (pH, gel, colonne et ses caractéristiques, volume du gel, tampon, type d'élution, volume résultat, durée, pertes inhérentes au tamis et pertes totales)
- * stocke ces informations en BD

- propose les techniques applicables après une chromatographie sur échangeurs d'ions en fonction des caractéristiques de l'échantillon :
- si volume résultat < 20 ml
 - alors une chromatographie est à nouveau applicable
 - sinon il faut concentrer avant de poursuivre
- une chromatographie est à nouveau applicable

→ si choix = électrophorèse
le SE ne possède pas d'informations permettant de guider le chercheur dans le choix du matériel nécessaire à l'application de cette technique; cependant il :

- demande à l'utilisateur d'introduire les valeurs de PI obtenues en résultat de l'électrophorèse
- calcule les pourcentages de contamination de la protéine à purifier par les contaminants
- affiche le menu des choix de techniques

→ si choix = récapitulatif
le SE :

- demande à l'utilisateur d'introduire la date pour laquelle il souhaite visualiser les informations relatives aux techniques exécutées ce jour-là
- affiche ces informations

→ si choix = quitter
le SE affiche un message

IV. CONCLUSION

IV. CONCLUSION

En guise de conclusion, nous nous permettrons de critiquer certains aspects de l'outil de développement VP-Expert. Dans un second temps, nous exposerons les problèmes rencontrés lors de la conception du logiciel. Enfin, nous proposerons quelques idées afin d'améliorer le travail que nous avons réalisé.

1. Critiques de l'outil

Parmi les critiques négatives que nous pouvons émettre, voici ce que nous pouvons dire :

- le formalisme utilisé dans VP-Expert, à savoir les règles "IF...THEN...ELSE", nous a fortement contraintes ; en effet, il n'est pas toujours possible de représenter les connaissances, et plus particulièrement l'enchaînement logique du programme, sous forme de règles de production. Ceci nous a conduites à utiliser un certain nombre de "trucs" afin d'y parvenir, et souvent, ceux-ci sont à l'origine d'un manque de rigueur au niveau de la programmation.
- comme il a été décrit précédemment, nous avons fréquemment travaillé sur base de la partie linéaire de courbes d'étalonnage de gels dont l'abscisse est exprimée en logarithme en base 10 du PM. Nous avons dû constater la pauvreté de VP-Expert pour ce qui concerne les fonctions mathématiques ; en effet, cet outil ne présente que le

logarithme népérien et non le logarithme en base 10.

Ceci nous a amenées à recourir à un facteur de conversion, ce qui peut constituer une source d'erreurs supplémentaires.

- le nombre de niveaux de récursion étant fixé à 20, nous avons parfois été confrontées lors de l'exécution au message suivant : "RECURSION TOO DEEP". Cette caractéristique limite fortement l'accès répété à une même règle ou à un groupe de règles, ce qui nous a souvent obligées à rendre le programme plus séquentiel, et par conséquent moins souple à l'utilisation, les possibilités de "bouclage" étant réduites.
- une remarque accessoire concerne l'affichage des données provenant d'une base de données. De fait, celui-ci n'apparaît pas d'une façon très esthétique.
- ainsi que nous l'avons signalé dans une section antérieure, l'interaction avec le tableur Lotus 1-2-3 a donné lieu à quelques problèmes. Il est effectivement impossible de calculer des valeurs au niveau de Lotus et de les manipuler au sein du programme, et ce durant la même consultation.
- pour ce qui est du manuel d'utilisation, outre la présentation des écrans, des différentes fonctions classées alphabétiquement, ... qui apparaissent clairement, il nous semble qu'il est parfois trop peu explicite et imprécis.

Par ailleurs, nous pouvons cependant relever des points positifs :

- l'interface utilisateur de VP-Expert est conviviale.
- notons encore la facilité avec laquelle VP-Expert interagit avec une base de données (contrairement à ce qu'il a été dit pour le tableur Lotus 1-2-3).

- en ce qui concerne l'apprentissage de l'outil, ce dernier s'effectue relativement rapidement ; ceci semble dû à la simplicité de la conception de l'outil (un seul type de chaînage, fonctionnalités élémentaires,...) et à la syntaxe "IF...THEN...ELSE...".

2. Critiques du travail

La phase d'acquisition des connaissances s'est faite relativement facilement, et ce étant donné notre formation de biologiste.

Notre programme offre des opportunités d'impression à différents endroits du logiciel.

Les informations stockées dans les fichiers de la base de données ne peuvent pas, dans l'état actuel du programme, être détruites.

Alors que l'ajout d'une protéine est pris en considération pour la suite de l'exécution du programme, les informations relatives à une nouvelle colonne et à un nouveau gel sont uniquement stockées, et ne constituent en rien une base supplémentaire pour le raisonnement du moteur d'inférence (les règles manipulant ces données ne sont pas générales car elles sont écrites en fonction des informations existant actuellement dans la base de données).

3. Perspectives d'avenir

Il nous semble nécessaire, lors de développements ultérieurs du système, d'actualiser et de permettre la manipulation par le logiciel des informations stockées dans le gestionnaire de bases de données. En effet, comme il a été dit précédemment, lorsqu'un nouveau gel ou une nouvelle colonne est introduite par l'utilisateur, les caractéristiques de ces derniers sont stockées, mais les règles susceptibles de manipuler ces nouvelles informations ne se créent pas "automatiquement". Une manière de résoudre cela serait, pour l'établissement de l'équation relative à la partie linéaire de la courbe d'étalonnage d'un nouveau gel et pour le calcul des PM limites, de stocker une seule équation qui serait générale pour le traitement de tous les gels. Il serait donc demander à l'utilisateur d'introduire les différents paramètres spécifiques à la courbe d'étalonnage du gel souhaité.

Une autre amélioration serait de pouvoir visualiser au moyen d'un graphe les PM limites d'une même protéine avec différents gels, qui seraient matérialisés par leur "droite" d'étalonnage.

Dans un même ordre d'idée, il ne serait pas inutile non plus de pouvoir détruire des informations stockées dans Dbase III : ceci contribuerait à éviter une croissance exagérée des fichiers, et permettrait donc une meilleure gestion des données. De fait, les opérateurs du laboratoire utilisent préférentiellement tel gel ou telle colonne plutôt que tel(le) autre. Ce phénomène relève de critères très peu formels ou d'habitudes qu'ils ont acquises au cours de longues années de pratique de laboratoire. Dès lors, rien ne sert de conserver dans la base de données ou ailleurs des informations dont ils n'auront que faire.

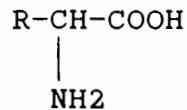
Un autre point susceptible d'être amélioré consiste en la conception d'une interface meilleure et encore plus conviviale. L'interface offerte par VP-Expert, loin d'être démunie, possède le désavantage d'être assez rigide et de ne pas accorder assez de liberté à son utilisateur. Nous avons tenté d'en augmenter l'ergonomie en permettant des impressions de tableaux, de récapitulatifs, et ce à divers endroits du programme. Cependant, le laboratoire de Biochimie Cellulaire, pour qui nous avons réalisé ce travail, est équipé d'environnements de type Macintosh, et serait particulièrement intéressé de posséder un programme similaire pouvant tourner sur cet ordinateur et offrant la convivialité bien connue de ce type de matériel (menus déroulants, souris,...).

Toutes les améliorations dont nous venons de parler ont été envisagées lors de la conception du programme, mais nombre d'entre elles n'ont pu être effectivement réalisées, et ce principalement à cause du caractère rigide de l'outil qu'est VP-Expert. En effet, un programme conçu avec ce type d'outil restreint, non seulement la liberté d'action du concepteur, mais encore celle de l'utilisateur.

GLOSSAIRES

GLOSSAIRE BIOLOGIQUE

ACIDE AMINE (= aminoacide) : molécule organique libérée par l'hydrolyse des protéines. Il en existe une vingtaine dans les protéines naturelles. La formule générale d'un acide aminé est :



ADSORPTION : pénétration superficielle d'un gaz ou d'un liquide dans un solide.

AMPHOLYTE : électrolyte possédant à la fois la fonction acide et la fonction basique.

CELLOPHANE : pellicule cellulosique transparente.

CENTRIFUGATION : séparation des constituants d'un mélange.

CHROMATOGRAMME : distribution de substances obtenue sur un support après chromatographie.

CHROMATOGRAPHIE : technique de séparation de composés chimiques, acides aminés par exemple, mettant à profit leur diffusion différentielle dans un support (papier, gel d'amidon, colonne de résines, etc.).

CONCENTRATION : (1) rapport de la masse d'une substance dissoute au volume de solution ; elle se mesure donc comme la masse spécifique en grammes par centimètre cube (ou par litre). (2) opération consistant à réduire le volume d'une solution par la chaleur, le vide, etc. de sorte que sa richesse augmente en la substance dissoute.

DENATURATION : changement apporté dans la structure d'une molécule, ce qui entraîne une perte de ses propriétés. La dénaturation peut être produite soit par effet thermique, soit par des procédés chimiques.

DESSALAGE : enlèvement de composants de faibles poids moléculaire (composants organiques et sels inorganiques) de ceux de poids moléculaire plus élevé.

DIALYSE : à partir d'une solution, méthode de séparation des grosses molécules colloïdales et des molécules plus petites (ions, sels,...) par diffusion à travers une membrane sélective.

DIELECTRIQUE : capable d'emmagasiner de l'énergie statique.

ECHANTILLON : prélèvement effectué sur une substance afin d'en faire l'analyse chimique ou d'y effectuer une mesure mécanique ou physique.

ELECTROPHORESE : technique de séparation d'acides aminés, de protéines, etc. reposant sur leur différence de charges électriques ou de taille moléculaire.

ELECTROLYTE : composé chimique qui, dissout, peut subir l'électrolyse, c'est-à-dire une décomposition lors du passage d'un courant électrique.

ELUTION : destruction par un liquide dissolvant convenable d'un produit d'adsorption, plus spécialement réalisé sur une colonne d'adsorption chromatographique ou sur une colonne d'échangeurs d'ions.

ENZYME : protéine métabolique intervenant dans toute réaction de dégradation ou de synthèse dans les cellules. Les enzymes sont des biocatalyseurs qui agissent en infime quantité et ne se trouvent pas modifiées au cours des réactions.

FORCE IONIQUE : force liée à la concentration en ions dans le milieu.

FRACTIONNEMENT : procédé d'analyse immédiate basé sur une différence de propriétés physiques (solubilité, fusion, distillation, adsorption).

GEL : solution rigide et élastique de nature colloïdale pouvant servir de support pour certaines chromatographies et électrophorèses.

GLUCOSE : hexose du groupe des aldohexoses universellement répandu dans les organismes vivants où il représente une source très importante d'énergie dans les mécanismes de la respiration cellulaire et de la fermentation.

GRADIENT : variation progressivement décroissante à partir d'un point maximal de la concentration d'une substance dans un environnement particulier.

HEMOGLOBINE : pigment des globules rouges du sang assurant le transport de l'oxygène et du gaz carbonique entre l'appareil respiratoire et les cellules de l'organisme.

ION : atome ou groupement d'atomes portant une charge électrique positive ou négative.

LIGAND : désigne toute molécule caractérisée par sa tendance à se lier à une autre.

LIMITE D'EXCLUSION : terme utilisé dans la technique du tamis moléculaire pour désigner le poids moléculaire limite en-dessous duquel les séparations de protéines sont possibles avec un gel particulier. Par contre, au-dessus de cette limite, les molécules sont incapables de pénétrer les pores du gel et seront donc éluées avec le volume interstitiel de la colonne.

LIQUIDE CEREBROSPINAL : liquide trouvé dans l'encéphale et la moëlle épinière.

MACROMOLECULE : grosse molécule (polymère), formée par association d'un grand nombre de molécules simples (monomères) et dont le poids moléculaire est supérieur à 1000.
Exemples : protéines, acides nucléiques,...

PEPTIDE : molécule constituée de l'union d'un petit nombre d'acides aminés.

pH : valeur changée de signe du logarithme décimal de la concentration d'une solution aqueuse en ions H⁺. Le pH se mesure comme une différence de potentiel ou à l'aide d'indicateurs colorés. Si le pH < 7, la solution est acide ; si le pH = 7, elle est neutre ; si le pH > 7, elle est alcaline.

PI : les acides aminés ont tous un pH défini auquel ils ne migreront vers aucune électrode sous des conditions électrophorétiques. Cette valeur de pH où la substance possède une charge globale neutre est connue sous le nom de PI, et est caractéristique pour chaque substance.

pK : expression signifiant $-\log K$,
avec $K = \frac{[A] * [B]}{[AB]}$

expression de la loi d'action des masses pour la dissociation d'une molécule AB. Si le corps (exemple l'acide sulfurique) subit deux dissociations successives on trouvera des expressions notées pK1 et pK2.

PLASMA : liquide où baigne les globules du sang et la lymphe.

PRECIPITATION : formation d'un corps insoluble dans un milieu de réaction donné.

PROTEINE : macromolécule constituée par une ou plusieurs chaînes peptidiques elles-mêmes formées d'une succession d'acides aminés unis par des liaisons peptidiques et dont la séquence est déterminée génétiquement. On peut distinguer d'un point de vue structural des protéines fibreuses dans les tissus de soutien (collagène, kératine) et des protéines globulaires possédant souvent une activité biologique (hormones, enzymes).

SACCHAROSE : sucre appartenant à la famille des glucides.

SOLUBILITE : propriété que possède une substance de se dissoudre dans un autre.

SOLUTE : corps dissout dans une solution.

SOLUTION : mélange homogène de substances dont l'une au moins est liquide.

SOLVANT : produit liquide qui a le pouvoir de dissoudre d'autres substances et que l'on utilise en chromatographie pour provoquer la séparation des composés d'un mélange.

SUBSTRAT : désigne la substance (ou le groupe de substances) sur laquelle se fixe une enzyme qui la transforme.

TAMPON : mélange de solutions où le pH ne s'altère pratiquement pas au cours d'une dilution. On prend généralement le mélange d'un acide fort (ou d'une base forte) avec un sel organique. Toute variation de pH y provoque une réaction tendant à restaurer l'équilibre primitif.

ULTRAFILTRATION : dialyse accompagnée de pression, réalisée à travers la paroi d'un ultrafiltre et destinée à empêcher l'afflux du solvant vers la solution et à expulser le liquide intermicellaire.

GLOSSAIRE INFORMATIQUE

ALGORITHME : procédure composée d'une suite d'instructions qui garantit de trouver un résultat correct. Pour développer un programme conventionnel, le programmeur spécifie l'algorithme que suivra le programme.

ATTRIBUT : propriété d'un objet. Par exemple, un pneu est un attribut d'une voiture. Les attributs peuvent avoir différentes valeurs, ainsi que des valeurs spécifiques dans des situations particulières.

BASE DE CONNAISSANCES : module du système expert qui contient les faits et heuristiques relatifs à un domaine. La connaissance peut prendre la forme de faits, règles ou objets.

CHAINAGE ARRIERE : stratégie de contrôle qui régule l'ordre dans lequel les inférences sont exécutées. Dans un système à base de règles, ce chaînage est engendré par l'objectif (ou but). Le système essaie de déterminer une valeur pour cet objectif ; il identifie les règles qui concluent à une valeur pour ce dernier, puis il essaie de déterminer si les clauses "IF..." de ces règles sont vérifiées. Par conséquent, les attributs des clauses "IF..." deviennent objectifs secondaires et le processus se répète.

CHAINAGE AVANT : stratégie de contrôle qui régule l'ordre dans lequel les inférences sont exécutées. Dans un système à base de règles, ce chaînage commence par passer en revue les faits connus et ensuite déduit toutes les règles dont les clauses "IF..." sont établies. Par après, le système commence un autre cycle et agit de même à partir des nouveaux faits établis, et ainsi de suite. Ce processus se répète jusqu'à ce que le programme atteigne un but ou ne parvienne plus à établir de nouvelles possibilités.

CONNAISSANCE : collection intégrée de faits et relations qui, quand elle est exécutée, produit des performances notables. La quantité et la qualité de la connaissance possédée par une personne ou un ordinateur peut être jugée par la variété des situations dans lesquelles la personne ou le programme peut successivement obtenir des résultats.

ENVIRONNEMENT DE PROGRAMMATION : concept se situant à mi-chemin entre un langage de programmation et un outil de développement. Un langage permet à l'utilisateur une complète flexibilité. Un outil contraint l'utilisateur de plusieurs manières. Un environnement de programmation, comme OPS5, fournit un certain nombre de routines pré-établies pouvant faciliter un développement rapide de programmes à base de règles.

EXPERTISE : habileté et connaissance que certains êtres humains possèdent et qui entraînent des performances bien au-dessus de la norme. L'expertise consiste fréquemment en des quantités massives d'informations couplées à des règles de "bonne pratique", simplifications, faits rares et procédures, tout ceci étant mélangé et permettant à l'expert d'analyser des problèmes spécifiques d'une manière efficace.

FACTEUR DE CONFIANCE : poids numérique attribué à un fait ou à une relation indiquant la confiance qu'a l'expert dans le fait ou la relation.

FAIT : énoncé dont la validité est acceptée. Dans la plupart des systèmes experts, un fait consiste en un attribut et en une ou plusieurs valeurs qui sont associées à cet attribut.

HEURISTIQUE : règle de "bonne pratique" ou tout autre moyen qui permet à son utilisateur de tirer des conclusions sans être certain. Contrairement aux algorithmes, les heuristiques ne garantissent pas de trouver une solution correcte.

INFERENCE : procédé par lequel de nouveaux faits sont dérivés de faits précédemment établis.

INGENIEUR DE CONNAISSANCES : individu dont la spécialité est de déterminer des problèmes, d'acquérir les connaissances et de construire les systèmes à base de connaissances. Ceci implique une formation dans les sciences du "savoir", dans le domaine informatique et plus particulièrement en intelligence artificielle. Ceci suggère également de l'expérience dans le développement d'un ou plusieurs systèmes experts.

INTELLIGENCE ARTIFICIELLE : domaine de l'informatique s'intéressant aux concepts et méthodes d'inférence symbolique et à la représentation symbolique de la connaissance utilisée dans ces inférences. Il s'agit d'une discipline académique comme la physique ; ce n'est pas un produit mais bien un vaste programme de recherche visant à établir ce dont les ordinateurs sont capables.

INTERFACE : lien entre un programme informatique et le monde extérieur. Les systèmes experts possèdent une interface développeur, une interface utilisateur et une interface système à travers lesquelles ils entrent en relation avec d'autres software ou hardware.

LANGAGE NATUREL : secteur d'activité de l'intelligence artificielle qui étudie les techniques permettant aux systèmes informatiques d'accepter des "entrées" et de fournir des "sorties" dans un langage conventionnel tel que l'anglais ou le français.

MAINTENANCE D'UN SYSTEME EXPERT : contrairement aux programmes conventionnels, rarement mis à jour, les systèmes experts, de par leur nature, sont très facilement modifiables. De plus, les systèmes conventionnels sont "complets", tandis que les systèmes experts doivent être constamment améliorés par l'addition de nouvelles règles.

MOTEUR D'INFERENCE : programme chargé, dans un système expert, d'activer les règles concernées par le problème à résoudre et d'y associer les éléments pertinents de la base de connaissances, soit par chaînage avant, soit par chaînage arrière. Quand un moteur d'inférence est séparé de la base de connaissances, il s'agit alors d'un outil de développement de système expert.

OUTIL : "package" informatique qui simplifie l'effort de développement dans la construction d'un système expert. La majorité des outils contiennent un moteur d'inférence, différentes interfaces utilisateur et des systèmes d'aide à l'acquisition des connaissances ; seule manque la base de connaissances.

PROTOTYPE : dans le développement des systèmes experts, un prototype est une version initiale du système qui est développée afin de tester l'efficacité globale de la représentation des connaissances et des stratégies d'inférence employées pour résoudre le problème.

RECHERCHE EN LARGEUR D'ABORD : stratégie de contrôle qui examine toutes les règles ou objets du même niveau de la hiérarchie avant d'examiner n'importe quelle règle du niveau inférieur.

RECHERCHE EN PROFONDEUR D'ABORD : stratégie de contrôle dans laquelle une règle ou un objet du plus haut niveau est examiné et, ensuite, ce sont les règles ou objets du niveau immédiatement inférieur qui le sont. En procédant de la sorte, la recherche se réalise vers le bas de l'arbre et ce jusqu'à ce que la fin soit atteinte.

RECHERCHE EXHAUSTIVE : stratégie de recherche qui examine systématiquement chaque chemin possible dans un arbre de décision ou dans un réseau. Cette méthode de recherche est coûteuse et/ou impossible pour de nombreux problèmes.

REGLE "IF...THEN..." : règle établissant une relation entre une série de faits dans une clause "IF..." et un ou plusieurs faits dans une clause "THEN...". Les règles peuvent être définitionnelles comme par exemple SI <feminin> ET <mariee> ALORS <femme> ou heuristiques comme la règle suivante : SI <pluvieux> ALORS <prendre parapluie>.

RESOLUTION DE PROBLEMES : processus dans lequel on part d'un état initial et on effectue une recherche à travers l'espace d'états afin d'identifier la séquence d'opérations (actions) qui permettront d'atteindre l'objectif fixé.

ROBOTIQUE : néologisme désignant l'ensemble des techniques permettant la conception et la mise en oeuvre de dispositifs destinés à se substituer à l'homme dans ses fonctions motrices et sensorielles avec un degré plus ou moins grand de capacité d'adaptation à l'environnement. Ces dispositifs peuvent soit être programmés pour réaliser des actions données, soit agir par apprentissage.

SYSTEME D'EXPLOITATION (= O.S.) : ensemble de programmes de base d'une machine permettant d'utiliser tous les services disponibles et assurant en particulier la gestion des travaux, les opérations d'entrée/sortie sur les périphériques, l'affectation des ressources aux différents processus, l'accès aux bibliothèques de programmes et aux fichiers, ainsi que la compatibilité des travaux. L'O.S. MS/DOS a été développé par la société Microsoft à partir de 1981 et choisi par IBM pour son ordinateur personnel (PC).

SYSTEME EXPERT : environnement logiciel et matériel permettant de résoudre des problèmes très spécifiques et bien délimités en utilisant une base de connaissances symbolique et un mécanisme de raisonnement (moteur d'inférence) travaillant sur la base de connaissances et les données du problème.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

A. DOMAINE BIOLOGIQUE

ALTMAN P.L. and DITTMER D.S., 1972, Biology Data Book (vol.1), Federation of American Societies for Experimental Biology, pp. 370-380.

BAVER H.H., CHRISTIAN G.D. and O'REILLY E., 1978, Instrumental Analysis, Allyn and bacon, Inc.

BERTHILLIER A., 1972, La chromatographie et ses applications, Dunod, Paris.

BOSCHETTI E., 1987, Biofutur : Le Technoscope : Chromatographie liquide, le passage obligé pour purifier des protéines produites par culture de cellules animales, n.14.

CROIX-ROUGE DE BELGIQUE, Département Central de Fractionnement, Bruxelles : données relatives aux protéines sanguines.

DENNEY R.C., 1982, A dictionary of chromatography, Mac Millan Reference Books.

DETERMAN H., 1969, Chromatographie sur gel : filtration, perméation, tamissage moléculaire, Manon & Company.

Encyclopedia of physical science and technology, 1987, Academic Press INC, vol.2, pp 768-770, et vol.4, pp 862-880.

Encyclopédie internationale des sciences et des techniques, 1971, Les Presses de la Cité, vol.3, pp 251 et vol.5, pp 23-27.

LEBRUN Y., LHERMITTE B., 1988, Développement d'un prototype de système expert pour la planification d'expériences en biochimie, Mémoire présenté en vue de l'obtention du titre de licencié et maître en informatique, F.U.N.D.P. Namur.

LEHNINGER A.L., 1977, Biochimie : Bases moléculaires de la structure et des fonctions cellulaires, Flammarion Medecine-Sciences.

MARSHALL A.G., 1978, Biophysical Chemistry : Principles, Techniques and Applications, John Wiley & Sons, Inc.

MARTIN D.W., MAYES P.A., RODWELL V.W., 1985, Précis de biochimie de Harper, Les presses de l'université Laval, ESKA.

OVELETTE G., 1977, La chromatographie, Guide scientifique en chimie-biochimie.

PHARMACIA FINE CHEMICALS, Sephadex - théorie et pratique de la
filtration sur gel.

REMACLE J., 1984, Cours de première licence en biochimie :
Techniques biochimiques, F.U.N.D.P., Namur.

SCHULTZE H.E. and HEREMANS J.F., 1966, Molecular biology of
human proteins, vol.1, Elsevier Publishing
Company.

B. DOMAINE INFORMATIQUE

- BARR A., FEIGENBAUM E., 1982, Applications-Oriented AI Research : Education in Hanbook of Artificial Intelligence, vol.2, Heuristech. Press & William Kaufman Inc.
- BARRETO J., AI Programming Pardigmes, pp. 34-38 et LISP, pp.39-41.
- BUCHANAN B.G., SHORTLIFFE E.H., 1984, Rule-based expert systems : The MYCIN experiments of the Stanford heuristic programming project, Addison Wesley Publishing Company.
- DUPUIS H., 1988, Intelligence Artificielle : Le marché des experts, Sciences et Technologie, Tendances, n.302,
- DUCHASTEL Ph., 1989, ICAI Systems : issues in computer tutoring, Computers Educ., vol.13, n.1, pp 95-100.
- GAILLY-GOFFAUX L., RIBBENS D., 1988, Réalisation d'un système expert, Gestion 2000 Management & Prospective, n.2, pp 109-136.
- HARMON P., MAUS R., MORRISSEY W., 1988, Expert Systems Tools and Applications, John Wiley & Sons.

HAYES-ROTH F., WATERMAN D.A., LENAT D.B., 1983, Building Experts systems, Addison Wesley Publishing Company, Inc.

HART P., 1986, IEEE Expert : Interview : Peter Hart talks about expert systems, pp 96-99.

Journal de réflexion sur l'informatique, n.9, Institut d'informatique, F.U.N.D.P.

KLAHR P., WATERMAN D.A.(éditeurs), 1986, Experts Systems, Techniques, Tools and Applications, Addison Wesley Publishing Company.

MICHIE D. (éditeur), 1979, Expert Systems in the microelectronic age, Edinburgh University Press.

Micro Data Base Systems, INC (éditeur), 1985, GURU, le système expert destiné à l'entreprise : manuel de référence (2 vol), version 1.

Nexpert-Object fundamentals, version 1.1, 1987-1988, manuel d'utilisation de Nexpert-Object.

O'KEEFE R.M., BALCIO O., SMITH E.P., 1987, Validating Expert System Performance, IEEE Expert, vol.2, n. 4, pp 81-90.

RICH E., 1983, Artificial Intelligence, Mc Graw-Hill, pp 102-106.

SAWYER B., HARMON P., 1986, Tools, Expert Systems Strategies,
vol.2, n.4.

SHEIL B. 1988, Réflexions sur l'intelligence artificielle,
Harvard-L'Expansion, n.48, pp 17-24.

ANNEXE


```

!*****
!***** ACTIONS *****
!*****
runtime;
execute;
endoff;
autoquery;
actions CCALL B:\JULES
        MENU proteine,ALL,fiprotri,nom
        var = 1
        boolhplc = 0
        find date_jour
        cls
        find tab1
        find tab2
        find message_1
        display " "
        find message_2
        cls
        display "A present, nous allons vous demander d'introduire quelques
caracteristiques specifiques au melange.
        "

        find technique_a
        find concentration
        cls
        find technique_2
        find testconst
        find suitetech_1;

!*****
!***** RULES *****
!*****
!----- affichage de la 'liste des constituants' -----
RULE 1
if var = 1
then tab1=1
        display "LISTE DES CONSTITUANTS :
veuillez parcourir soigneusement la liste ci-dessous et noter si oui ou non
la proteine que vous voulez purifier s'y trouve
        ";
!----- boucle de l'affichage des constituants jusqu'a 15 -----
RULE 2
if var = 1
then tab2 = OK
        g = 1
        whileknown nom
                get all,fiprotri,nom
                display "                {nom}"
                g = (g+1)
                reset liste
                find liste
        end;

RULE 3
if g > 15
then liste = KO
        display "                Pour continuer : RETURN~"
        cls
        reset g
        reset tab2
        find tab2
else liste = OK;

```

!-----introduction d'un nouveau constituant dans la BD -----

RULE 4

if infos = non

then message_1 = displayed

display "Veuillez dans ce cas entrer les valeurs des differents parametres
qui vous

sont proposes.Si vous ignorez une valeur, entrez '0'."

find nom

find nature

find pm

find pi

find bi_stab

find bs_stab

find bi_prec

find bs_prec

find pour_stab

cls

reset showliste

find showliste

display "

Pour la suite du programme, il est necessaire que certains parametres qui
vous sont demandes ci-dessus aient une valeur non nulle.

Des lors, des tests vont maintenant etre effectues, et, eventuellement, il
sera necessaire de reintroduire certaines valeurs.~"

find test

append fiprotri

cls

find showliste;

!----- verification des valeurs de PM,PI,POUR-STAB,BI-STAB,BS-STAB -----

RULE 5

if PM = 0

then test = negatif

reset PM

find PM

find test2

else test = continuer

find test2;

RULE 6

if PI = 0

then test2 = negatif

reset PI

find PI

find test3

else test2 = continuer

find test3;

RULE 7

if pour_stab = 0

then test3 = negatif

reset pour_stab

find pour_stab

find test4

else test3 = continuer

find test4;

RULE 8

if bi_stab = 0 or

bs_stab = 0

then test4 = negatif

```

        reset bi_stab
        reset bs_stab
        find bi_stab
        find bs_stab
    else test4 = continuer;

!----- affichage des caracteristiques de la proteine choisie -----
RULE 9
if infos = oui
then message_1 = displayed
    cls
    find proteine
    close fiprotri
    get proteine = nom,fiprotri,all
    cls
    reset showliste
    find showliste;

RULE 10
if var = 1
then afficher = OK
    reset showliste
    find showliste
    reset test
    find test;

!----- affichage proprement dit de la liste des caracteristiques -----
RULE 11
if var = 1
then showliste = OK
    display "La proteine {nom} a les caracteristiques suivantes :
        nature = {nature}
        PM = {pm} (daltons)
        PI = {pi} (pH)
        borne inferieure de stabilite = {bi_stab} (pH)
        borne superieure de stabilite = {bs_stab} (pH)
        borne inferieure de precipitation = {bi_prec} (%)
        borne superieure de precipitation = {bs_prec} (%)
        pourcentage de denaturation = {pour_stab} (%/jour)
        ~";

!----- modification de la nature de la proteine-----
RULE 12
if infos2 = oui and
    choix = nature
then message_2 = displayed
    close fiprotri
    get proteine = nom,fiprotri,all
    reset nature
    find nature
    reset afficher
    find afficher
    PUT fiprotri
    cls
    reset choix
    find choix
    reset message_2
    find message_2;

!----- modification du P.M. de la proteine -----
RULE 13
if infos2 = oui and

```

```

    choix = pm
then message_2 = displayed
    close fiprotri
    get proteine = nom,fiprotri,all
    reset pm
    find pm
    reset afficher
    find afficher
    PUT fiprotri
    cls
    reset choix
    find choix
    reset message_2
    find message_2;

```

!----- modification du P.I. de la proteine -----

```

RULE 14
if infos2 = oui and
    choix = p1
then message_2 = displayed
    close fiprotri
    get proteine = nom,fiprotri,all
    reset p1
    find p1
    reset afficher
    find afficher
    PUT fiprotri
    cls
    reset choix
    find choix
    reset message_2
    find message_2;

```

!----- modification de la borne inferieure de stabilite de la proteine -----

```

RULE 15
if infos2 = oui and
    choix = bi_stab
then message_2 = displayed
    close fiprotri
    get proteine = nom,fiprotri,all
    reset bi_stab
    find bi_stab
    reset afficher
    find afficher
    PUT fiprotri
    cls
    reset choix
    find choix
    reset message_2
    find message_2;

```

!-----modification de la borne superieure de stabilite de la proteine-----

```

RULE 16
if infos2 = oui and
    choix = bs_stab
then message_2 = displayed
    close fiprotri
    get proteine = nom,fiprotri,all
    reset bs_stab
    find bs_stab
    reset afficher
    find afficher

```

```
PUT fiprotri
cls
reset choix
find choix
reset message_2
find message_2;
```

!----- modification de la borne inferieure de precipitation de la proteine-----

```
RULE 17
if infos2 = oui and
  choix = bi_prec
then message_2 = displayed
  close fiprotri
  get proteine = nom,fiprotri,all
  reset bi_prec
  find bi_prec
  reset afficher
  find afficher
  PUT fiprotri
  cls
  reset choix
  find choix
  reset message_2
  find message_2;
```

!----- modification de la borne superieure de precipitation de la proteine-----

```
RULE 18
if infos2 = oui and
  choix = bs_prec
then message_2 = displayed
  close fiprotri
  get proteine = nom,fiprotri,all
  reset bs_prec
  find bs_prec
  reset afficher
  find afficher
  PUT fiprotri
  cls
  reset choix
  find choix
  reset message_2
  find message_2;
```

!----- modification du pourcentage de denaturation de la proteine -----

```
RULE 19
if infos2 = oui and
  choix = pour_stab
then message_2 = displayed
  close fiprotri
  get proteine = nom,fiprotri,all
  reset pour_stab
  find pour_stab
  reset afficher
  find afficher
  PUT fiprotri
  cls
  reset choix
  find choix
  reset message_2
  find message_2;
```

!----- modification d'aucune caracteristique de la proteine -----

```

RULE 20
if infos2 = oui and
    choix = aucune
then message_2 = displayed
    cls;

!----- modification d'aucune caracteristique de la proteine -----
RULE 21
if infos2 = non
then message_2 = displayed
    display "Des lors,nous allons elaborer un plan de purification a partir de
ces
caracteristiques.          ~"

    cls;

!----- choix technique de depart (precipitation) -----
RULE 22
if vol_ech > 18
then technique_a = PRECIPITATION_ISOLECTRIQUE
    display "Nous vous conseillons d'appliquer une {technique_a}."
    reset vol_res
    reset quant_res
    find vol_res
    vol_ech = (vol_res)
    find quant_res
    quant_ech = (quant_res)
    reset technique_a
    find technique_a;

RULE 23
if vol_ech <= 18 and
    texture = proteine_soluble
then technique_a = suite_et_purif
    reset quant_res;

RULE 24
if vol_ech <= 18 and
    texture = proteine_membranaire
then technique_a = PRECIPITATION_ISOLECTRIQUE
    display "Nous vous conseillons d'appliquer une {technique_a}."
    reset vol_res
    reset quant_res
    reset texture
    find vol_res
    vol_ech = (vol_res)
    find quant_res
    quant_ech = (quant_res)
    reset technique_a
    find technique_a;

!----- calcul de la concentration initiale -----
RULE 25
if vol_ech <> 0 and
    quant_ech <> 0
then concentration = (quant_ech/vol_ech)
    format concentration,5.2
    display "La concentration de depart est de {concentration} mg/ml.
~"

else concentration =incorrecte
    reset vol_ech
    reset quant_ech

```

```

        reset concentration
        display "Vous avez commis une erreur ; veuillez reintroduire les valeurs
suiivantes : "
        find concentration;

```

```

!----- determination des P.I. par electrofocalisation (ou non) -----
RULE 26
if piconta = non
then technique_2 = trouve
    display "Il est dans ce cas necessaire d'appliquer une ELECTROFOCALISATION
afin de pouvoir determiner les PI des contaminants de la proteine a purifier.
Ensuite, nous allons vous demander d'introduire les differentes valeurs de PI
que vous avez obtenues en resultat de l'application de la technique.
Si vous ne connaissez plus aucune valeur de PI, tapez '?'."

```

```

        find respi
        cls
    else technique_2 = trouve
        cls
        display "Dans ce cas, entrez TOUTES les valeurs de PI des contaminants.
Si vous ne connaissez plus aucune valeur de PI, tapez '?'."
        find respi;

```

```

!----- saisie des differentes valeurs de P.I. -----
RULE 27
if technique_2 = trouve
then respi = OK
    i = 1
    find contapi
    vecpi[i] = (contapi)
    whileknown contapi
        reset contapi
        i = (i+1)
        find contapi
        vecpi[i] = (contapi)
    end
    maxpi = (i-1)
    cls
    find calc2
    cls
    find technique_3;

```

```

!----- determination des P.M. par HPLC (ou non) -----
!----- IMPLEMENTATION DE L'HPLC -----
RULE 28
if pmconta = non
then technique_3 = trouve
    boolhplc = 1
    display "Il est dans ce cas necessaire d'appliquer une HPLC afin
de pouvoir determiner les PM des contaminants de la proteine a purifier."
    find vargell
    cls
    display "Le gel etant determine, nous allons vous demander de choisir parmi
le
materiel dont vous disposez au laboratoire une colonne pour effectuer votre
HPLC. Une fois que vous l'aurez determinee, nous vous demanderons d'introdui-
re le volume du gel, ainsi que le volume d'echantillon a déposer au sommet de
la colonne (ces caracteristiques sont a lire sur la feuille technique de la
colonne)."

```

```

find colhplc
find volgelhplc
find voldephplc
find volreshplc
concenthplc = (concentration)
format concenthplc,5.2
find dureehplc
find tamphplc
reset stabilite
reset stabhre
reset pertech
reset dureetech
reset pertetot
reset inhtech
dureetech = (dureehplc * (1/60))
format dureetech,5.2
inhtech = 10
find stabilite
cls
display "EN RESUME, nous vous conseillons : une HPLC avec

```

```

gel = {affgel}
colonne = {colhplc}
tampon = {tamponhplc}
concentration avant l'application de l'HPLC = {concenthplc} (mg/ml)
volume resultant de l'HPLC (vol.estimate) = {volreshplc} (ml)
duree de l'HPLC (duree estimee) = {dureehplc} (min)

```

La perte totale due a l'application de la technique et a la denaturation de la proteine est de {pertetot} (%).

```

~"
reset impression
reset imprimer1
find imprimer1
res[1] = (volreshplc)
res[2] = (tamponhplc)
res[3] = (pertetot)
res[4] = (concenthplc)
nomtech = hplc
gel = (affgel)
volgel = (volgelhplc)
colonne = (colhplc)
diamcol = 0
tampon = (tamponhplc)
ph = 0
typelut = -
durtech = (dureetech)
inhtech = 10
volres = (volreshplc)
concent = (concenthplc)
pertegl = (pertetot)
date = (date_jour)
append fichres
cls

```

display "Maintenant, nous vous demandons d'introduire les differentes valeurs

rs
de PM des contaminants que vous avez obtenues en resultat de l'application de la technique.

Quand vous aurez introduit toutes vos valeurs, entrez '?'

```

"
find respm
find calc

```



```

else technique_3 = trouve
  display "Dans ce cas, entrez TOUTES les valeurs de PM des contaminants.
  Quand vous aurez introduit toutes vos valeurs, entrez '?'
  "

  find respm
  pertetot = 0
  res[1] = (vol_ech)
  res[3] = 0
  res[4] = (concentration)
  cls;

!----- test pour l'impression du resume de l'HPLC-----
RULE 29
if impression = oui
then imprimer1 = OK
  display "Veuillez des lors vous assurer que la connection avec l'imprimante
  est etablie et que cette derniere est 'ON LINE'.
  ~"

  pdisplay "EN RESUME, nous vous conseillons : une HPLC avec

  gel = {affgel}
  colonne = {colhplc}
  tampon = {tamponhplc}
  concentration avant l'application de l'HPLC = {concenthplc} (mg/ml)
  volume resultant de l'HPLC (vol.estimate) = {volreshplc} (ml)
  duree de l'HPLC (duree estimee) = {dureehplc} (min)

  La perte totale due a l'application de la technique et a la denatu-
  ration de la proteine est de {pertetot} (%).
  ~"

else imprimer1 = KO;

!----- saisie des differentes valeurs de P.M. -----
RULE 30
if technique_3 = trouve
then respm = OK
  i = 1
  find contapm
  vecpm[i] = (contapm)
  whileknown contapm
    reset contapm
    i = (i+1)
    find contapm
    vecpm[i] = (contapm)
  end
  maxpm = (i-1);

!----- CALCUL DES P.M. LIMITES POUR DIFFERENTS GELS -----
!----- gel G75 : calcul des PM limites -----
RULE 31
if var = 1
then vargell = apparu
  reset y
  reset y1
  reset y2
  find fourchette
  y = (2.8674-(0.2549*(@LOG(PM))))
  y1 = (y+(fourchette))
  y2 = (y-(fourchette))
  PMLIM1 = (@EXP(-(y1-2.8674)/0.2549))
  PMLIM2 = (@EXP(-(y2-2.8674)/0.2549))
  cls

```

```

reset testkav1
find testkav1
find vargel2;

```

```

RULE 32
if y1 < 0
then testkav1 = OK
    PMLIM1 = 0
    PMLIM2 = 0
    display "Avec un SEPHADEX G75 :

```

Le poids moleculaire de la proteine se situe en dehors des limites de separation de ce gel; celui-ci n'est donc d'aucune utilite dans le cas present.

```

"
else testkav1 = continuer
    reset testkav2
    find testkav2;

```

```

RULE 33
if y2 < 0
then testkav2 = OK
    PMLIM2 = 0
    display "Avec un SEPHADEX G75 :

```

Vu le PM de la proteine, ce gel ne separera que les contaminants dont le PM est inferieur a {PM} daltons.

```

"
else testkav2 = KO
    format PMLIM1,7.0
    format PMLIM2,7.0
    display "Avec un SEPHADEX G75 :

```

Toutes les proteines de poids moleculaire inferieur a {PMLIM1} daltons ou superieur a {PMLIM2} daltons seront eliminees; les proteines dont le PM est compris entre ces deux bornes contamineront la proteine que vous desirez purifier, et qui elle possede un poids moleculaire de {PM} daltons.

```

";
!----- gel G100 : calcul des PM limites -----

```

```

RULE 34
if var = 1
then vargel2 = apparu
    reset y
    reset y1
    reset y2
    y = (3.0735-(0.2609*(@LOG(PM))))
    y1 = (y+(fourchette))
    y2 = (y-(fourchette))
    PMLIM3 = (@EXP(-(y1-3.0735)/0.2609))
    PMLIM4 = (@EXP(-(y2-3.0735)/0.2609))
    reset testkav3
    find testkav3
    cls
    find vargel3;

```

```

RULE 35
if y1 < 0
then testkav3 = OK
    PMLIM3 = 0

```

```
PMLIM4 = 0
display "Avec un SEPHADEX G100 :
```

Le poids moleculaire de la proteine se situe en dehors des limites de separation de ce gel; celui-ci n'est donc d'aucune utilite dans le cas present.

RETURN

~"

```
else testkav3 = continuer
  reset testkav4
  find testkav4;
```

```
RULE 36
if y2 < 0
then testkav4 = OK
  PMLIM4 = 0
  display "Avec un SEPHADEX G100 :
```

Vu le PM de la proteine, ce gel ne separera que les contaminants dont le PM est inferieur a {PM} daltons.

RETURN

~"

```
else testkav4 = KO
  format PMLIM3,7.0
  format PMLIM4,7.0
  display "Avec un SEPHADEX G100 :
```

Toutes les proteines de poids moleculaire inferieur a {PMLIM3} daltons ou superieur a {PMLIM4} daltons seront eliminees; les proteines dont le PM est compris entre ces deux bornes contamineront la proteine que vous desirez purifier, et qui elle possede un poids moleculaire de {PM} daltons.

RETURN

~";

!-----gel SEPH.6B. : calcul des PM limites-----

```
RULE 37
if var = 1
then vargel3 = apparu
  reset y
  reset y1
  reset y2
  y = (2.1690-(0.1515*(@LOG(PM))))
  y1 = (y+(fourchette))
  y2 = (y-(fourchette))
  PMLIM5 = (@EXP(-(y1-2.1690)/0.1515))
  PMLIM6 = (@EXP(-(y2-2.1690)/0.1515))
  reset testkav5
  find testkav5
  find vargel4;
```

RULE 38

```

if y1 < 0
then testkav5 = OK
    PMLIM5 = 0
    PMLIM6 = 0
    display "Avec un SEPHAROSE 6B :

```

Le poids moleculaire de la proteine se situe en dehors des limites de separation de ce gel; celui-ci n'est donc d'aucune utilite dans le cas present.

```

"
else testkav5 = continuer
    reset testkav6
    find testkav6;

```

```

RULE 39
if y2 < 0
then testkav6 = OK
    PMLIM6 = 0
    display "Avec un SEPHAROSE 6B :

```

Vu le PM de la proteine, ce gel ne separera que les contaminants dont le PM est inferieur a {PM} daltons.

```

"
else testkav6 = KO
    format PMLIM5,7.0
    format PMLIM6,7.0
    display "Avec un SEPHAROSE 6B :

```

Toutes les proteines de poids moleculaire inferieur a {PMLIM5} daltons ou superieur a {PMLIM6} daltons seront eliminees; les proteines dont le PM est compris entre ces deux bornes contamineront la proteine que vous desirez purifier, et qui elle possede un poids moleculaire de {PM} daltons.

";

!-----gel SEPH.S200 : calcul des PM limites-----

```

RULE 40
if var = 1
then vargel4 = apparu
    reset y
    reset y1
    reset y2
    y = (2.57-(0.208*(@LOG(PM))))
    y1 = (y+(fourchette))
    y2 = (y-(fourchette))
    PMLIM7 = (@EXP(-(y1-2.57)/0.208))
    PMLIM8 = (@EXP(-(y2-2.57)/0.208))
    reset testkav7
    find testkav7
    cls
    find vargel5;

```

```

RULE 41
if y1 < 0
then testkav7 = OK
    PMLIM7 = 0
    PMLIM8 = 0
    display "Avec un SEPHACRYL S200 :

```

Le poids moleculaire de la proteine se situe en dehors des limites de

separation de ce gel; celui-ci n'est donc d'aucune utilite dans le cas present.

RETURN

~"

```
else testkav7 = continuer
  reset testkav8
  find testkav8;
```

```
RULE 42
if y2 < 0
then testkav8 = OK
  PMLIM8 = 0
  display "Avec un SEPHACRYL S200 :
```

Vu le PM de la proteine, ce gel ne separera que les contaminants dont le PM est inferieur a {PM} daltons.

RETURN

~"

```
else testkav8 = KO
  format PMLIM7,7.0
  format PMLIM8,7.0
  display "Avec un SEPHACRYL S200 :
```

Toutes les proteines de poids moleculaire inferieur a {PMLIM7} daltons ou superieur a {PMLIM8} daltons seront eliminees; les proteines dont le PM est compris entre ces deux bornes contamineront la proteine que vous desirez purifier, et qui elle possede un poids moleculaire de {PM} daltons.

RETURN

~";

!----- gel ULTRO.ACA.44 : calcul des PM limites -----

```
RULE 43
if var = 1
then vargel5 = apparu
  reset y
  reset y1
  reset y2
  y = (3.134-(0.261*(@LOG(PM))))
  y1 = (y+(fourchette))
  y2 = (y-(fourchette))
  PMLIM9 = (@EXP(-(y1-3.134)/0.261))
  PMLIM10 = (@EXP(-(y2-3.134)/0.261))
  reset testkav9
  find testkav9
  find vargel6;
```

```
RULE 44
if y1 < 0
then testkav9 = OK
  PMLIM9 = 0
  PMLIM10 = 0
```

display "Avec un ULTROGEL ACA 44 :

Le poids moleculaire de la proteine se situe en dehors des limites de separation de ce gel; celui-ci n'est donc d'aucune utilite dans le cas present.

"

```
else testkav9 = continuer
  reset testkav10
  find testkav10;
```

RULE 45

```
if y2 < 0
then testkav10 = OK
  PMLIM10 = 0
  display "Avec un ULTROGEL ACA 44 :
```

Vu le PM de la proteine, ce gel ne separera que les contaminants dont le PM est inferieur a {PM} daltons.

"

```
else testkav10 = KO
  format PMLIM9,7.0
  format PMLIM10,7.0
  display "Avec un ULTROGEL ACA 44 :
```

Toutes les proteines de poids moleculaire inferieur a {PMLIM9} daltons ou superieur a {PMLIM10} daltons seront eliminees; les proteines dont le PM est compris entre ces deux bornes contamineront la proteine que vous desirez purifier, et qui elle possede un poids moleculaire de {PM} daltons.

";

!----- gel ULTRO.ACA.22 : calcul des PM limites -----
!----- affichage du tableau recapitulatif -----

RULE 46

```
if var = 1
then vargel6 = apparu
  reset y
  reset y1
  reset y2
  y = (3.879-(0.273*(@LOG(PM))))
  y1 = (y+(fourchette))
  y2 = (y-(fourchette))
  PMLIM11 = (@EXP(-(y1-3.879)/0.273))
  PMLIM12 = (@EXP(-(y2-3.879)/0.273))
  reset testkav11
  find testkav11
  cls
  format PMLIM1,7.0
  format PMLIM2,7.0
  format PMLIM3,7.0
  format PMLIM4,7.0
  format PMLIM5,7.0
  format PMLIM6,7.0
  format PMLIM7,7.0
  format PMLIM8,7.0
  format PMLIM9,7.0
  format PMLIM10,7.0
  format PMLIM11,7.0
  format PMLIM12,7.0
  display_."
```

TABLEAU RECAPITULATIF : (valeurs exprimees en daltons)

NOM du gel	PM limites calcules	PM de la proteine	limites de fractionnement
Sephadex G75	{PMLIM1} - {PMLIM2}	{PM}	6000 - 65000
Sephadex G100	{PMLIM3} - {PMLIM4}	{PM}	1000 - 100000
Sepharose 6B	{PMLIM5} - {PMLIM6}	{PM}	10000 - 3000000
Sephacryl S200	{PMLIM7} - {PMLIM8}	{PM}	5000 - 150000
Ultrogel ACA 44	{PMLIM9} - {PMLIM10}	{PM}	10000 - 140000
Ultrogel ACA 22	{PMLIM11} - {PMLIM12}	{PM}	100000 - 1200000

display "Les valeurs de PM calculees nulles correspondent a des PM limites se situant hors de la partie lineaire de la courbe d'etalonnage."

```

reset impression
reset imprimer2
find imprimer2
find quelgell
find affgel
find satisgel;

```

!----- test de l'impression du tableau des gels (PMLIM) pour l'HPLC-----

RULE 47

```

if impression = oui
then imprimer2 = OK

```

display "Veuillez des lors vous assurer que la connection avec l'imprimante est etablie et que cette derniere est 'ON LINE'."

pdisplay "

TABLEAU RECAPITULATIF : (valeurs exprimees en daltons)

NOM du gel	PM limites calcules	PM de la proteine	limites de fractionnement
Sephadex G75	{PMLIM1} - {PMLIM2}	{PM}	6000 - 65000
Sephadex G100	{PMLIM3} - {PMLIM4}	{PM}	1000 - 100000
Sepharose 6B	{PMLIM5} - {PMLIM6}	{PM}	10000 - 3000000
Sephacryl S200	{PMLIM7} - {PMLIM8}	{PM}	5000 - 150000
Ultrogel ACA 44	{PMLIM9} - {PMLIM10}	{PM}	10000 - 140000
Ultrogel ACA 22	{PMLIM11} - {PMLIM12}	{PM}	100000 - 1200000

pdisplay "Les valeurs de PM calculees nulles correspondent a des PM limites se situant hors de la partie lineaire de la courbe d'etalonnage."

```

else imprimer2 = KO;

```

RULE 48

```

if y1 < 0

```

```

then testkav11 = OK

```

```

PMLIM11 = 0

```

```

PMLIM12 = 0

```

```

display "Avec un ULTROGEL ACA 22 :

```

Le poids moleculaire de la proteine se situe en dehors des limites de separation de ce gel; celui-ci n'est donc d'aucune utilite dans le cas present.

RETURN

```

~"
else testkav11 = continuer
  reset testkav12
  find testkav12;

```

```

RULE 49
if y2 < 0
then testkav12 = OK
  PMLIM12 = 0
  display "Avec un ULTROGEL ACA 22 :

```

Vu le PM de la proteine, ce gel ne separera que les contaminants dont le PM est inferieur a {PM} daltons.

RETURN

```

~"
else testkav12 = KO
  format PMLIM11,7.0
  format PMLIM12,7.0
  display "Avec un ULTROGEL ACA 22 :

```

Toutes les proteines de poids moleculaire inferieur a {PMLIM11} daltons ou superieur a {PMLIM12} daltons seront eliminees; les proteines dont le PM est compris entre ces deux bornes contamineront la proteine que vous desirez purifier, et qui elle possede un poids moleculaire de {PM} daltons.

RETURN

!----- DETERMINATION DU GEL LE PLUS ADEQUAT -----

```

RULE 50
if PMLIM1 >= 6000 and
  PMLIM1 <= 65000 and
  PMLIM2 >= 6000 and
  PMLIM2 <= 65000 and
  PMLIM1 <> 0 and
  PMLIM2 <> 0
then quelgel1 = OK
  diff[1] = ((PMLIM2)-(PMLIM1))
  find mindiff
else quelgel1 = KO
  diff[1] = 0
  find quelgel2;

```

```

RULE 51
if PMLIM3 >= 1000 and
  PMLIM3 <= 100000 and
  PMLIM4 >= 1000 and
  PMLIM4 <= 100000 and
  PMLIM3 <> 0 and
  PMLIM4 <> 0
then quelgel2 = OK
  diff[2] = ((PMLIM4)-(PMLIM3))
  find mindiff
else quelgel2 = KO
  diff[2] = 0
  find quelgel3;

```


RULE 52

```

if PMLIM5 >= 10000 and
  PMLIM5 <= 3000000 and
  PMLIM6 >= 10000 and
  PMLIM6 <= 3000000 and
  PMLIM5 <> 0 and
  PMLIM6 <> 0
then quelgel3 = OK
  diff[3] = ((PMLIM6)-(PMLIM5))
  find mindiff
else quelgel3 = KO
  diff[3] = 0
  find quelgel4;

```

RULE 53

```

if PMLIM7 >= 5000 and
  PMLIM7 <= 150000 and
  PMLIM8 >= 5000 and
  PMLIM8 <= 150000 and
  PMLIM7 <> 0 and
  PMLIM8 <> 0
then quelgel4 = OK
  diff[4] = ((PMLIM8)-(PMLIM7))
  find mindiff
else quelgel4 = KO
  diff[4] = 0
  find quelgel5;

```

RULE 54

```

if PMLIM9 >= 10000 and
  PMLIM9 <= 140000 and
  PMLIM10 >= 10000 and
  PMLIM10 <= 140000 and
  PMLIM9 <> 0 and
  PMLIM10 <> 0
then quelgel5 = OK
  diff[5] = ((PMLIM10)-(PMLIM9))
  find mindiff
else quelgel5 = KO
  diff[5] = 0
  find quelgel6;

```

RULE 55

```

if PMLIM11 >= 100000 and
  PMLIM11 <= 1200000 and
  PMLIM12 >= 100000 and
  PMLIM12 <= 1200000 and
  PMLIM11 <> 0 and
  PMLIM12 <> 0
then quelgel6 = OK
  diff[6] = ((PMLIM12)-(PMLIM11))
  find mindiff
else quelgel6 = KO
  diff[6] = 0;

```

RULE 56

```

if var = 1
then mindiff = OK
  min = 999999
  z = 1
  whileknown diff[z]

```

```

        reset minimum
        find minimum
        z = (z+1)
    end;

RULE 57
if diff[z] <> 0 and
    diff[z] <= (min)
then minimum = trouve
    min = (diff[z])
    indice = (z)
else minimum = KO
    min = 999999
    indice = 7;

!----- AFFICHAGE DU GEL LE PLUS ADEQUAT -----
RULE 58
if indice = 1
then affgel = SEPHADEX_G75
    display "Le gel le plus approprié est le SEPHADEX G75.
";

RULE 59
if indice = 2
then affgel = SEPHADEX_G100
    display "Le gel le plus approprié est le SEPHADEX G100.
";

RULE 60
if indice = 3
then affgel = SEPHAROSE_6B
    display "Le gel le plus approprié est le SEPHAROSE 6B.
";

RULE 61
if indice = 4
then affgel = SEPHACRYL_S200
    display "Le gel le plus approprié est le SEPHACRYL S200.
";

RULE 62
if indice = 5
then affgel = ULTROGEL_ACA_44
    display "Le gel le plus approprié est l'ULTROGEL_ACA_44.
";

RULE 63
if indice = 6
then affgel = ULTROGEL_ACA_22
    display "Le gel le plus approprié est l'ULTROGEL_ACA_22.
";

RULE 64
if indice = 7
then affgel = inconnu
    display "Etant donné les gels que nous avons stockés dans la base de données,
et surtout les critères sur lesquels nous nous basons pour déterminer celui qui
est le plus adéquat, il ne nous est pas possible de vous conseiller un gel.
";

!----- degré de satisfaction du choix du gel -----
!----- affichage des gels disponibles -----
RULE 65

```

```

if satisf = oui
then satisfgel = OK
else satisfgel = KO
  cls
  close fichgelt
  display "Dans ce cas voici les gels disponibles :
  CARACTERISTIQUES DES GELS :

```

```

-----
      NOM du                MARQUE                limites de
      gel                                fractionnement (d)
-----
      whileknown type
      get all,fichgelt,all
      display
" {type}                {marque}                {domfract1}-{domfracts} "
      end
      display
"-----
      find nouvgel
      reset affgel
      affgel = (satisfgel2);

```

```

RULE 66
if satisfgel2 = autre
then nouvgel = OK
  display "Veuillez dans ce cas introduire les differents parametres qui vous
sont proposes. Si vous ignorez une valeur, tapez '0'."
  find type
  find marque
  find domfract1
  find domfracts
  find coeffns
  append fichgelt
  satisfgel2 = (type)
else nouvgel = KO;

```

```

!----- CHOIX DES TAMPONS LES PLUS APPROPRIES POUR HPLC -----
RULE 67
if pi <= 7.2
then tamphplc = OK
  tamponhplc = phosphate
  reset tamphplc;

```

```

RULE 68
if pi < 8.2 and
  pi > 7.2
then tamphplc = OK
  tamphplc[1] = tris
  tamphplc[2] = phosphate
  display "Les tampons utilisables sont : "
  reset l
  l = 1
  whileknown tamphplc[l]
    display "          {tamphplc[l]}"
    l = (l+1)
  end
  find choixstamp
  tamponhplc = (choixstamp)
  reset tamphplc;

```

```

RULE 69
if pi >= 8.2 .

```

```

then tamphplc = OK
    tamponhplc = tris
else tamphplc = KO;

!----- calcul du degre de contamination des proteines contaminantes -----
RULE 70
if var = 1
then calcul = true
    cls
    i = 1
    whileknown vecpm[i]
        reset calcul
        find calcul
        i = (i+1)
    end
    find tableau;

RULE 71
if vecpm[i] <= (PM)
then calcul = dr_gauche
    reset yconta[i]
    reset PMLIMX
    reset vrailim
    find vrailim
    yconta[i] = ((100/(PM-PMLIMX))*vecpm[i] - ((100*PMLIMX)/(PM-PMLIMX)))
    format yconta[i],6.2
    reset testneg2
    find testneg2
    display "- Le pourcentage de contamination de la proteine de PM {vecpm[i]}
daltons dans le melange sera de {yconta[i]} (%).\"
    reset proportion
    reset proptotal[i]
    find proportion
    proptotal[i] = ((proportion/100) * (yconta[i]))
    format proptotal[i],6.2
    display "Le pourcentage global de contamination de cette proteine dans le
melange sera de {proptotal[i]} (%).\"
    ~\"
else calcul = dr_droite
    reset yconta[i]
    reset PMLIMY
    reset vrailim
    find vrailim
    yconta[i] = ((100/(PM-PMLIMY))*vecpm[i] - ((100*PMLIMY)/(PM-PMLIMY)))
    format yconta[i],6.2
    reset testneg2
    find testneg2
    display "- Le pourcentage de contamination de la proteine de PM {vecpm[i]}
daltons dans le melange sera de {yconta[i]} (%).\"
    reset proportion
    reset proptotal[i]
    find proportion
    proptotal[i] = ((proportion/100) * (yconta[i]))
    format proptotal[i],6.2
    display "Le pourcentage global de contamination de cette proteine dans le
melange sera de {proptotal[i]} (%).\"
    ~\";

!----- test des pourcentages negatifs et remise a zero -----
RULE 72
if yconta[i] <= 0
then testneg2 = OK

```

```
        yconta[i] = 0.00
else testneg2 = KO;
```

```
!----- DETERMINATION DES VERITABLES P.M. LIMITES -----
```

```
RULE 73
if affgel = sephadex_G75
then vrailim = OK
    PMLIMX = (PMLIM1)
    PMLIMY = (PMLIM2);
```

```
RULE 74
if affgel = sephadex_G100
then vrailim = OK
    PMLIMX = (PMLIM3)
    PMLIMY = (PMLIM4);
```

```
RULE 75
if affgel = sepharose_6B
then vrailim = OK
    PMLIMX = (PMLIM5)
    PMLIMY = (PMLIM6);
```

```
RULE 76
if affgel = sephacryl_S200
then vrailim = OK
    PMLIMX = (PMLIM7)
    PMLIMY = (PMLIM8);
```

```
RULE 77
if affgel = ultrogel_aca_44
then vrailim = OK
    PMLIMX = (PMLIM9)
    PMLIMY = (PMLIM10)
else vrailim = KO;
```

```
RULE 78
if affgel = ultrogel_aca_22
then vrailim = OK
    PMLIMX = (PMLIM11)
    PMLIMY = (PMLIM12)
else vrailim = KO;
```

```
!----- calcul du volume resultat pour HPLC -----
```

```
RULE 79
if var = 1
then volreshplc = ((5*(volgelhplc/100)) + voldephplc)
    format volreshplc,6.2;
```

```
!----- calcul de la duree de HPLC -----
```

```
RULE 80
if debithplc <> 0
then dureehplc = ((volgelhplc + voldephplc)/debithplc)
    format dureehplc,5.1
else dureehplc = incorrecte
    reset dureehplc
    reset debithplc
    display "Vous avez commis une erreur : veuillez reintroduire la valeur
    suivante : "
    find dureehplc;
```

```
!----- affichage du recapitulatif de contamination des PM -----
```

```
RULE 81
```

```

if var = 1
then tableau = OK
  CLS
  display "
Le gel choisi etant {affgel}, les limites de PM calculees sont {PMLIMX}
et {PMLIMY}. Veuillez donc considerer les eventuels pourcentages nuls comme
etant des valeurs correspondant a des contaminants totalement exclus du gel.

```

RECAPITULATION DES POURCENTAGES DE CONTAMINATION :

CONTAMINANTS	PM (d)	POURCENTAGE (%)	POURC.TOTAL (%)

i = 1			
whileknown vecpm[i]			
display			
"contaminant {i}	{vecpm[i]}	{yconta[i]}	{p
roptotal[i]}	~"		
i = (i+1)			
end			
display "			
----- fin du tableau -----			
~"			
reset impression			
reset imprimer3			
find imprimer3			
cls;			

!----- test de l'impression du recapitulatif de contamination des PM-----

RULE 82

```

if impression = oui
then imprimer3 = OK
  display "Veuillez des lors vous assurer que la connection avec l'imprimante
est etablie et que cette derniere est 'ON LINE'.
~"

```

```

pdisplay "
Le gel choisi etant {affgel}, les limites de PM calculees sont {PMLIMX}
et {PMLIMY}. Veuillez donc considerer les eventuels pourcentages nuls comme
etant des valeurs correspondant a des contaminants totalement exclus du gel.

```

RECAPITULATION DES POURCENTAGES DE CONTAMINATION :

CONTAMINANTS	PM (d)	POURCENTAGE (%)	POURC.TOTAL (%)

i = 1			
whileknown vecpm[i]			
pdisplay			
"contaminant {i}	{vecpm[i]}	{yconta[i]}	{p
roptotal[i]}	~"		
i = (i+1)			
end			
pdisplay "			
----- fin du tableau -----			
~"			
else imprimer3 = KO;			

!----- verification du nombre total de constituants -----

RULE 83

```

if nbreconst <> 0 and
  nbreconst = (maxpm + 1)
then testconst = OK
else testconst = KO
  reset nbreconst

```

```

reset testconst
display "Vous avez commis une erreur."
find testconst;

!----- orientation vers les techniques applicables -----
RULE 84
if nbreconst <= 15    and
   res[1] >= 0.5      and
   res[1] <= 18
then suitetech_1 = intermediaire
   concentid = 10
   cls
   find suitetech_2
   find suitetech_3
   find suitetech_4
   find suitetech_5
   display " "
   find applitech
else suitetech_1 = KO
   display "Le nombre de constituants par unite de volume est trop important
pour appliquer une chromatographie.
Nous vous conseillons de DILUER votre echantillon.~"
   reset vol_res
   find vol_res
   res[1] = (vol_res)
   reset suitetech_1
   find suitetech_1;

!----- TESTS DE LA CONCENTRATION PAR RAPPORT A LA CONCENTRATION IDEALE -----
RULE 85
if concentration > (concentid)
then suitetech_2 = tamis_dilution
   display "Après avoir analyse ces informations, nous vous conseillons :

- le TAMIS MOLECULAIRE (si vous estimez que votre concentration qui est de
{res[4]} mg/ml est proche de 10 mg/ml, la concentration ideale)
Toutefois, du fait de son pouvoir separateur relativement faible, de sa
lenteur et d'un aspect peu productif, nous vous conseillons d'appliquer
cette technique dans les etapes finales de la purification, moment ou la
proteine a separer est faiblement contaminee.
- DILUER l'echantillon pour ameliorer la concentration (si vous estimez que
le volume de votre echantillon de {vol_ech} ml est proche de 18 ml)";

RULE 86
if concentration = (concentid)
then suitetech_2 = tamis_moleculaire
   display "Vous pouvez appliquer le {suitetech_2}.";

RULE 87
if concentration < (concentid)
then suitetech_2 = tamis_concent
   display "Après avoir analyse ces informations, nous vous conseillons :

- le TAMIS MOLECULAIRE
Toutefois, du fait de son pouvoir separateur relativement faible, de sa
lenteur et d'un aspect peu productif, nous vous conseillons d'appliquer
cette technique dans les etapes finales de la purification, moment ou la
proteine a separer est faiblement contaminee.
- CONCENTRER l'echantillon pour ameliorer la concentration qui est de
{res[4]} mg/ml, et l'amener proche de la concentration ideale,
i.e. 10 mg/ml.";
```

```

!-----comparaison des PI pour E-IONS -----
RULE 88
if var = 1
then suitetech_3 = OK
  reussite = OK
  i = 1
  whileknown vecpi[i]
    reset comparaison
    find comparaison
    i = (i+1)
  end
  find fintest;

!----- test de comparaison des PI pour E-I -----
RULE 89
if vecpi[i] <= (pi-1) or
  vecpi[i] >= (pi+1)
then comparaison = OK
else comparaison = KO
  reset reussite
  reussite = KO;

RULE 90
if reussite = OK
then fintest = OK
  display " - une CHROMATOGRAPHIE SUR ECHANGEURS D'IONS "
else fintest = KO
  display " - pas de CHROMATOGRAPHIE SUR ECHANGEURS D'IONS etant donne que l
e PI
  de la proteine a purifier n'est pas suffisamment different des PI des
  contaminants (difference necessaire = 1 unite de pH)";

!----- comparaison des PI pour ELECTROPHORESE -----
RULE 91
if var = 1
then suitetech_4 = OK
  reussite2 = OK
  c = 1
  whileknown vecpi[c]
    reset comparaison2
    find comparaison2
    c = (c+1)
  end
  find fintest2;

!----- test de comparaison des PI pour ELECTROPHORESE -----
RULE 92
if vecpi[c] <= (pi-0.5) or
  vecpi[c] >= (pi+0.5)
then comparaison2 = OK
else comparaison2 = KO
  reset reussite2
  reussite2 = KO;

RULE 93
if reussite2 = OK
then fintest2 = OK
  find testquant
else fintest2 = KO
  display " - pas d'ELECTROPHORESE etant donne que le PI de la proteine a pu
rifier
  n'est pas suffisamment different des PI des contaminants

```



```

(difference necessaire = 0.5 unite de pH)";

RULE 94
if quant_ech <= 1 and
  quant_ech >= 10
then testquant = OK
  display " - une ELECTROPHORESE"
else testquant = KO
  display " - pas d'ELECTROPHORESE etant donne que la quantite de l'echantil
lon
  (exprimee en mg) est inadequate.";

!----- comparaison des PI pour ELECTROFOCALISATION -----
RULE 95
if var = 1
then suitetech_5 = OK
  reussite3 = OK
  d = 1
  whileknown vecpi[d]
    reset comparaison3
    find comparaison3
    d = (d+1)
  end
  find fintest3;

!----- test de comparaison des PI pour ELECTROFOCALISATION -----
RULE 96
if vecpi[d] <= (pi-0.1) or
  vecpi[d] >= (pi+0.1)
then comparaison3 = OK
else comparaison3 = KO
  reset reussite3
  reussite3 = KO;

RULE 97
if reussite3 = OK
then fintest3 = OK
  display " - une ELECTROFOCALISATION "
else fintest3 = KO
  display " - pas d'ELECTROFOCALISATION etant donne que le PI de la proteine
a purifier
  n'est pas suffisamment different des PI des contaminants
  (difference necessaire = 0.1 unite de pH)";

!----- IMPLEMENTATION DE LA CHROMATO-IONS -----
RULE 98
if choixtech = chromato_ion
then applitech = implemente
  cls
  reset phion
  find phion
  reset volgelion
  reset quantdepion
  find volgelion
  volechion = (res[1])
  conction = (quantdepion/volechion)
  format conction,5.2
  reset tamponion
  k = 0
  find tamponion
  reset diamcolion
  find diamcolion

```

```

reset typeelution
find typeelution
reset volresion
find volresion
reset stabilite
reset stabhre
reset pertech
reset dureetech
reset pertetot
reset inhstech
dureetech = 4
inhstech = 15
find stabilite
pertecum = (pertetot + res[3])
format pertecum,5.2
cls
display "EN RESUME, nous vous conseillons : une CHROMATO-ION avec

```

```

pH = {pHion} (pH)
gel = {gelion}
diametre de la colonne = {diamcolion} (cm)
volume du gel = {volgelion} (ml)
tampon = {tamponion}
type d'elution possible = {typelution}
volume resultat = {volresion} (ml)
duree = {dureetech} (heures)

```

La perte totale due a l'application de la technique et a la denaturation de la proteine est de {pertetot} (%).
 Les pertes resultant de l'application d'autre(s) technique(s) s'elevent a {pertecum} (%).

~"

```

reset impression
reset imprimer4
find imprimer4
res[1] = (volresion)
res[2] = (tamponion)
res[3] = (pertecum)
res[4] = (concentration)
nomtech = echangeur_ion
gel = (gelion)
volgel = (volgelion)
colonne = -
diamcol = (diamcolion)
tampon = (tamponion)
ph = (phion)
typelut = (typelution)
durtech = (dureetech)
inhstech = 15
volres = (volresion)
concent = (concentration)
pertegl = (pertetot)
date = (date_jour)
append fichres
cls

```

```

display "Le volume obtenu etant de {res[1]} (ml), la concentration etant de {res[4]} (mg/ml) et le tampon utilise le {res[2]}, nous vous conseillons :

```

"

```

reset apresion
find apresion
display "

```

- une CHROMATOGRAPHIE SUR ECHANGEURS D'IONS

Cette technique presente la propriete de concentrer l'echantillon et ainsi est souvent utilisee dans les premiers stades de purification.
ATTENTION : verifiez la concentration en sels, et si besoin est, dialysez !
 De plus, si vous passez d'un echangeur d'anions a un echangeur de cations, et inversement, veillez au choix du tampon.

```

reset applitech
reset choixtech
find applitech;

```

!----- test de l'impression du resume de la CHROMA-IONS -----

RULE 99

```

if impression = oui
then imprimer4 = OK
  display "Veuillez des lors vous assurer que la connection avec l'imprimante
est etablie et que cette derniere est 'ON LINE'."
  ~"

```

pdisplay "EN RESUME, nous vous conseillons : une CHROMATO-ION avec

```

pH = {pHion} (pH)
gel = {gelion}
diametre de la colonne = {diamcolion} (cm)
volume du gel = {volgelion} (ml)
tampon = {tamponion}
type d'elution possible = {typelution}
volume resultat = {volresion} (ml)
duree = {dureetech} (heures)

```

La perte totale due a l'application de la technique et a la denaturation de la proteine est de {pertetot} (%).
 Les pertes resultant de l'application d'autre(s) technique(s) s'elevent a {pertecum} (%).

~"

```

else imprimer4 = KO;

```

!----- que faire apres une chromato-ions ? -----

RULE 100

```

if res[1] <= 20
then apresion = OK
  display "- un TAMIS MOLECULAIRE etant donne que le volume est inferieur a
20 (ml). Toutefois, du fait de son pouvoir separateur relativement faible, de
sa lenteur et d'un aspect peu productif, nous vous conseillons d'appliquer
cette technique dans les etapes finales de la purification, moment ou la pro-
teine a separer est accompagnee d'un nombre limite de contaminants faiblement
concentres."
else apresion = KO
  display "- de concentrer votre echantillon etant donne que le volume est su-
perieur a
20 (ml), ceci dans le cas ou vous souhaiteriez appliquer un TAMIS
MOLECULAIRE. De plus, nous vous indiquons que cette technique est surtout
utilisee dans les etapes finales de la purification, moment ou la proteine a
separer est accompagnee d'un nombre limite de contaminants faiblement
concentres.";

```

!-calcul degre de contam. pour PI (ions, electrophorese, electrofocalisation)

RULE 101

```

if var = 1
then calc2 = true
  cls
  i = 1
  whileknown vecpi[i]

```

```

        reset calcul2
        find calcul2
        i = (i+1)
    end
    reset tableau2
    find tableau2;

RULE 102
if vecpi[i] < (pi)
then calcul2 = dr_gauche
    find fourchette2
    reset containfsup
    reset y2conta[i]
    containfsup = (PI-fourchette2)
    y2conta[i] = ((100/(PI-containfsup))*vecpi[i] - ((100*containfsup)/(PI-containfsup)))
    format y2conta[i],6.2
    reset testneg
    find testneg
    display "- Le pourcentage de contamination de la proteine de PI {vecpi[i]}
pH dans le melange sera de {y2conta[i]} (%).
~"
else calcul2 = dr_droite
    find fourchette2
    reset containfsup
    reset y2conta[i]
    containfsup = (PI+fourchette2)
    y2conta[i] = ((100/(PI-containfsup))*vecpi[i] - ((100*containfsup)/(PI-containfsup)))
    format y2conta[i],6.2
    reset testneg
    find testneg
    display "- Le pourcentage de contamination de la proteine de PI {vecpi[i]}
pH dans le melange sera de {y2conta[i]} (%).
~";

!----- test des pourcentages negatifs et remise a zero -----
RULE 103
if y2conta[i] <= 0
then testneg = OK
    y2conta[i] = 0.00
else testneg = KO;

!-etude des degres de contam. selon un autre coefficient de resolution du gel--

RULE 104
if autrefourch2 = oui
then modifourch2 = OK
    reset fourchette2
    find fourchette2
    reset autrefourch2
    reset calc2
    find calc2
else modifourch2 = KO;

!----- affichage du tableau recapitulatif des pourcentages de contamination --
RULE 105
if var = 1
then tableau2 = OK
    cls
    display "
RECAPITULATION DES POURCENTAGES DE CONTAMINATION :

```

```

-----
CONTAMINANTS                PI (pH)                POURCENTAGE (%)
-----
      i = 1
      while known vecpi[i]
        display
"contaminant {i}                {vecpi[i]}                {y2conta[i]}
      }
        i = (i+1)
      end
      display "
----- fin du tableau -----
~"

      reset impression
      reset imprimer5
      find imprimer5
      reset modifourch2
      find modifourch2
      cls;

!----- test de l'impression du recapitulatif de contamination des PI-----
RULE 106
if impression = oui
then imprimer5 = OK
  display "Veuillez des lors vous assurer que la connection avec l'imprimante
est etablie et que cette derniere est 'ON LINE'.
  ~"
  pdisplay "
RECAPITULATION DES POURCENTAGES DE CONTAMINATION :
-----
CONTAMINANTS                PI (pH)                POURCENTAGE (%)
-----
      i = 1
      while known vecpi[i]
        pdisplay
"contaminant {i}                {vecpi[i]}                {y2conta[i]}
      }
        i = (i+1)
      end
      pdisplay "
----- fin du tableau -----
~"

else imprimer5 = KO;

!-----determination de pH, gel (IONS)-----

RULE 107
if pi <= ((bi_stab+bs_stab)/2)
then phion = (bs_stab)
  find gelion
else phion = (bi_stab)
  find gelion;

RULE 108
if phion > (pi)
then gelion = DEAE_SEPHADEX_A50
  display "Le gel le plus approprie est le {gelion}.
  ";

RULE 109
if phion < (pi) and
  phion <= .5

```

```

then gelion = SP_SEPHADEX_C50
  display "Le gel le plus approprié est le {gelion}.
";

```

```

RULE 110
if phion < (pi) and
  phion > 5
then gelion = CM_SEPHADEX_C50
  display "Le gel le plus approprié est le {gelion}.
";

```

```

!-----determination du diam_colonne (IONS)-----
RULE 111
if hautcolion <> 0 and
  res[1] <> 0
then diamcolion = (2*(@sqrt(volgelion/(hautcolion*3.14))))
  format diamcolion,5.2
  display "Le diamètre de la colonne est de l'ordre de {diamcolion} (cm).
";

```

```

!-----determination du volume du gel (IONS)-----
RULE 112
if quantdepion <> 0
then volgelion = ((quantdepion*(3.75/3))*18)
  format volgelion,6.2
  display "Le volume du gel est de {volgelion} (ml).
";

```

```

!-----choix du tampon (IONS)-----
RULE 113
if phion > 3.75 and
  phion < 5.75
then tamponion = OK
  k = (k+1)
  tampon[k] = ACIDE_ACETIQUE
  reset tamponion;

```

```

RULE 114
if phion > 6.20 and
  phion < 8.20
then tamponion = OK
  k = (k+1)
  tampon[k] = PHOSPHATE
  reset tamponion;

```

```

RULE 115
if phion > 2.1 and
  phion < 4.1
then tamponion = OK
  k = (k+1)
  tampon[k] = CITRATE_1
  reset tamponion;

```

```

RULE 116
if phion > 3.7 and
  phion < 5.7
then tamponion = OK
  k = (k+1)
  tampon[k] = CITRATE_2
  reset tamponion;

```

```

RULE 117

```

```

if phion > 4.4 and
  phion < 6.4
then tamponion = OK
  k = (k+1)
  tampon[k] = CITRATE_3
  reset tamponion;

```

```

RULE 118
if phion > 5.2 and
  phion < 7.2
then tamponion = OK
  k = (k+1)
  tampon[k] = CACODILATE
  reset tamponion;

```

```

RULE 119
if phion > 8.9 and
  phion < 10.9
then tamponion = OK
  k = (k+1)
  tampon[k] = GLYCINE
  reset tamponion;

```

```

RULE 120
if phion > 7.2 and
  phion < 9.2
then tamponion = OK
  k = (k+1)
  tampon[k] = TRIS_HCL
  find affichtamp
else tamponion = KO
  find affichtamp;

```

```

RULE 121
if k = 2
then affichtamp = OK
  display "Les tampons utilisables sont : "
  reset k
  k = 1
  whileknown tampon[k]
    display "          {tampon[k]}"
    reset citrate
    find citrate
    k = (k+1)
  end
  reset choixamp
  reset tamponion
  find choixamp
  tamponion = (choixamp)
else affichtamp = KO
  reset tamponion
  tamponion = (tampon[1]);

```

```

RULE 122
if tampon[k] = citrate_1
then citrate = OK
  display "Le pK de ce tampon est de 3.10.
          ";

```

```

RULE 123
if tampon[k] = citrate_2
then citrate = OK

```

```
display "Le pK de ce tampon est de 4.70.  
";
```

```
RULE 124  
if tampon[k] = citrate_3  
then citrate = OK  
display "Le pK de ce tampon est de 5.40."  
  
else citrate = KO;
```

```
!-----determination du type d'elution (IONS)-----
```

```
RULE 125  
if bi_stab <= (pi) and  
pi <= (bs_stab)  
then typeelution = OK  
typeelution[1] = pH  
typeelution[2] = FORCE_IONIQUE  
display "Les types d'elution possibles sont : "  
reset m  
m = 1  
whileknown typeelution[m]  
display " {typeelution[m]}"  
m = (m+1)  
end  
find choixtypeelution  
typeelution = (choixtypeelution)  
else typeelution = KO  
typeelution = FORCE_IONIQUE;
```

```
!-----determination du volume resultat (IONS) -----
```

```
RULE 126  
if decrochage = etapes  
then volresion = (0.5*(volgelion))  
format volresion,6.2  
display "Le volume de l'echantillon resultat est de {volresion} (ml).  
~"  
  
else volresion = (volgelion)  
display "Le volume de l'echantillon resultat est de {volresion} (ml).  
~";
```

```
!-----determination du % de denaturation pour chaque tech.-----
```

```
RULE 127  
if pour_stab <> 0  
then stabilite = OK  
stabhre = (pour_stab/24)  
pertech = (stabhre * dureetech)  
pertetot = (pertech + inhstech)  
format pertetot,5.2;
```

```
!-----IMPLEMENTATION DU TAMIS -----
```

```
RULE 128  
if choixtech = tamis_moleculaire  
then applitech = implemente  
cls  
reset fourchette  
reset autrefourch  
reset modifourch  
reset y  
reset y1  
reset y2  
reset vargel1  
reset vargel2
```



```

reset vargel3
reset vargel4
reset vargel5
reset vargel6
reset PMLIM1
reset PMLIM2
reset PMLIM3
reset PMLIM4
reset PMLIM5
reset PMLIM6
reset PMLIM7
reset PMLIM8
reset PMLIM9
reset PMLIM10
reset PMLIM11
reset PMLIM12
reset satisgel
reset satisf
reset diff[1]
reset diff[2]
reset diff[3]
reset diff[4]
reset diff[5]
reset diff[6]
reset quelgel1
reset quelgel2
reset quelgel3
reset quelgel4
reset quelgel5
reset quelgel6
reset mindiff
reset affgel
find vargel1
cls

```

display "Le gel etant determine, nous allons vous proposer une colonne
et un tampon pour effectuer votre TAMIS."

```

volechtamis = (res[1])
reset varcolOK2
reset varcolOK
reset numcol
reset colOK2
reset colOK
reset coltamis
find coltamis
reset volgeltamis
find volgeltamis
reset quantdephplc
find quantdephplc
quantdeptamis = (quantdephplc)
concentttamis = (quantdeptamis/volechtamis)
format concentttamis,5.2
reset volrestamis
find volrestamis
reset tamptamis
find tamptamis
reset stabilite
reset stabhre
reset pertech
reset dureetech
reset pertetot
reset inhtech
dureetech = 15

```

```

inhtech = 10
find stabilite
pertecum = (pertetot + res[3])
format pertecum,5.2
cls
display "EN RESUME, nous vous conseillons : un TAMIS MOLECULAIRE avec

gel = {affgel}
caracteristiques de la colonne :
    diametre = {diametre} (cm)
    hauteur = {hauteur} (cm)
    vitesse = {vitcolhplc} (ml/h)
    volume max. d'echantillon = {volmaxhplc} (ml)
tampon = {tamponsamis}
volume resultant du TAMIS = {volrestamis} (ml)
duree approximative du TAMIS = {dureetech} (heures)

La perte totale d'activite due a l'application d'un tamis molecu-
laire est de {pertetot} (%).
Les pertes resultant de l'application d'autre(s) technique(s) s'elevent
a {pertecum} (%)
~"

reset impression
reset imprimer6
find imprimer6
res[1] = (volrestamis)
res[2] = (tamponsamis)
res[3] = (pertecum)
res[4] = (concenttamis)
nomtech = tamis_moleculaire
gel = (affgel)
volgel = (volgeltamis)
colonne = -
diamcol = (diametre)
tampon = (tamponsamis)
ph = 0
typelut = 0
durtech = (dureetech)
inhtech = 10
volres = (volrestamis)
concent = (concenttamis)
pertegl = (pertetot)
date = (date_jour)
append fichres
cls
reset YNhp1c
find YNhp1c
cls
display "Le volume obtenu etant de {res[1]} (ml), la concentration etant de
{res[4]} (mg/ml) et le tampon utilise le/la {res[2]}, nous vous conseillons :
"

reset aprestamis
find aprestamis
reset suitetech_6
find suitetech_6
reset applitech
reset choixtech
find applitech;

!----- test de l'impression du resume du tamis moleculaire -----
RULE 129
if impression = oui

```

```

then imprimer6 = OK
  display "Veuillez des lors vous assurer que la connection avec l'imprimante
est etablie et que cette derniere est 'ON LINE'.
  ~"

```

```

pdisplay "EN RESUME, nous vous conseillons : un TAMIS MOLECULAIRE avec

```

```

gel = {affgel}
caracteristiques de la colonne :
  diametre = {diametre} (cm)
  hauteur = {hauteur} (cm)
  vitesse = {vitcolhplc} (ml/h)
  volume max. d'echantillon = {volmaxhplc} (ml)
tampon = {tampontamis}
volume resultant du TAMIS = {volrestamis} (ml)
duree approximative du TAMIS = {dureetech} (heures)

```

```

La perte totale d'activite due a l'application d'un tamis molecu-
laire est de {pertetot} (%).
Les pertes resultant de l'application d'autre(s) technique(s) s'elevent
a {pertecum} (%)
  ~"

```

```

else imprimer6 = KO;

```

```

!----- teste passage eventuel dans l'HPLC -----
RULE 130

```

```

if boolhplc = 1
then YNhplc = yes

```

```

  cls
  display "

```

```

Voici la liste des differentes valeurs de PM que vous aviez obtenues a la
sortie de l'HPLC que vous avez precedemment realisee :

```

```

  i = 1
  whileknown vecpm[i]
    display "          {vecpm[i]}"
    i = (i+1)
  end

```

```

  reset OKvalpm
  find OKvalpm

```

```

else YNhplc = no
  display "Maintenant, nous vous demandons d'introduire les differentes valeu
rs

```

```

de PM des contaminants que vous avez obtenues en resultat de l'application de
la technique.

```

```

Quand vous aurez introduit toutes vos valeurs, entrez '?'.
  "

```

```

  reset respm
  reset contapm
  o = 1
  whileknown vecpm[o]
    reset vecpm[o]
    o = (o+1)
  end
  reset maxpm
  reset calc
  reset tableau
  find respm
  find calc;

```

```

RULE 131
if OKvalpm2 = oui
then OKvalpm = inchange
  reset calc

```

```

    find calc
else OKvalpm = chgt
    display "Des lors, nous vous demandons d'introduire toutes les valeurs de P
M
que vous possédez; si vous ne connaissez plus aucune valeur, entrez '?'."

```

```

    reset respm
    reset contapm
    o = 1
    whileknown vecpm[o]
        reset vecpm[o]
        o = (o+1)
    end
    reset maxpm
    reset calc
    reset tableau
    find respm
    find calc;

```

```

RULE 132
lf var = 1
then suitetech_6 = OK
    reussitel = OK
    i = 1
    whileknown vecpi[i]
        reset comparaison1
        find comparaison1
        i = (i+1)
    end
    find fintest1;

```

```

RULE 133
lf vecpi[i] <= (pi-1) or
    vecpi[i] >= (pi+1)
then comparaison1 = OK
else comparaison1 = KO
    reset reussitel
    reussitel = KO;

```

```

RULE 134
lf reussitel = OK
then fintest1 = OK
    display " - une CHROMATOGRAPHIE SUR ECHANGEURS D'IONS.
    Cette technique presente la propriete de concentrer l'echantillon et ainsi
    est souvent utilisee dans les premiers stades de purification.
    ATTENTION : veuillez a ce que le tampon soit compatible avec celui que vous
    venez d'employer, ou mieux identique a celui-ci.
    De plus, verifiez la concentration en sels: si besoin est, dialysez !
"
else fintest1 = KO
    display " - pas de CHROMATOGRAPHIE SUR ECHANGEURS D'IONS etant donne que l
    PI
    le la proteine a purifier n'est pas suffisamment different des PI des conta-
    minants (difference necessaire = 1 unite de pH)
";

```

-----determination du tampon (TAMIS)-----

```

RULE 135
lf pi <= 7.2
then tamptamis = OK
    tampontamis = phosphate
    reset tamptamis;

```

```

RULE 136
if p1 < 8.2 and
  p1 > 7.2
then tamptamis = OK
  tamptamis[1] = tris
  tamptamis[2] = phosphate
  display "Les tampons utilisables sont : "
  reset 1
  l = 1
  while known tamptamis[l]
    display "      {tamptamis[l]}"
    l = (l+1)
  end
  find choixamp
  tamponamis = (choixamp)
  reset tamptamis;

RULE 137
if p1 >= 8.2
then tamptamis = OK
  tamponamis = tris
else tamptamis = KO;

!----- calcul du volume resultat (TAMIS)-----
RULE 138
if var = 1
then volrestamis = ((5*(volgeltamis/100)) + volechtamis)
  format volrestamis,6.2;

!----- choix de la colonne (TAMIS) -----
RULE 139
if volechtamis <= .25
then coltamis = 1
  find numcol;

RULE 140
if volechtamis <= 1
then coltamis = 2
  find numcol;

RULE 141
if volechtamis <= 3.5
then coltamis = 3
  find numcol;

RULE 142
if volechtamis <= 20
then coltamis = 4
  find numcol
else coltamis = KO
  display "Le volume de votre echantillon etant trop important, il faut le
  concentrer !~";

!-----affichage de la colonne conseilllee-----
RULE 143
if var = 1
then numcol = trouve
  close fichcolt
  get coltamis = numero,fichcolt,all
  display "Nous vous conseillons une COLONNE dont les caracteristiques sont :
    diametre = {diametre} (cm)

```

```

    hauteur = {hauteur}    (cm)
    vitesse = {vitesse}    (ml/h)
    volume max. d'echantillon = {volmax}    (ml)

```

Cette colonne a ete selectionnee car elle est la plus adequate etant donne le volume de l'echantillon.

```

    diamcolhplc = (diametre)
    hautcolhplc = (hauteur)
    vitcolhplc = (vitesse)
    volmaxhplc = (volmax)
    reset satisf
    find varcolOK;

```

!-----affichage des autres colonnes disponibles-----

RULE 144

```

if satisf = oui
then varcolOK = OK
else varcolOK = KO
  cls
  close fichcolt
  display "Dans ce cas, voici les colonnes disponibles :
  CARACTERISTIQUES DES COLONNES :

```

NUMERO	DIAMETRE (cm)	HAUTEUR (cm)	VITESSE (ml/hre)	VOL.MAX. (ml)
numrec = 0				
whileknown numero				
get all,fichcolt,all				
numrec = (numrec + 1)				
display				
" {numero}	{diametre}	{hauteur}	{vitesse}	
{volmax}	"			
end				
display				
find varcolOK2;				

!-----choix d'une autre colonne-----

RULE 145

```

if colOK2 = 1
then varcolOK2 = OK
  reset coltamis
  coltamis = 1
  reset numero
  reset diametre
  reset hauteur
  reset vitesse
  reset volmax
  close fichcolt
  get coltamis = numero,fichcolt,all
  numcolhplc = (numero)
  diamcolhplc = (diametre)
  hautcolhplc = (hauteur)
  vitcolhplc = (vitesse)
  volmaxhplc = (volmax);

```

RULE 146

```

if colOK2 = 2
then varcolOK2 = OK
  reset coltamis

```

```

coltamis = 2
reset numero
reset diametre
reset hauteur
reset vitesse
reset volmax
close fichcolt
get coltamis = numero, fichcolt, all
numcolhplc = (numero)
diamcolhplc = (diametre)
hautcolhplc = (hauteur)
vitcolhplc = (vitesse)
volmaxhplc = (volmax);

```

```

RULE 147
if colOK2 = 3
then varcolOK2 = OK
    reset coltamis
    coltamis = 3
    reset numero
    reset diametre
    reset hauteur
    reset vitesse
    reset volmax
    close fichcolt
    get coltamis = numero, fichcolt, all
    numcolhplc = (numero)
    diamcolhplc = (diametre)
    hautcolhplc = (hauteur)
    vitcolhplc = (vitesse)
    volmaxhplc = (volmax);

```

```

RULE 148
if colOK2 = 4
then varcolOK2 = OK
    reset coltamis
    coltamis = 4
    reset numero
    reset diametre
    reset hauteur
    reset vitesse
    reset volmax
    close fichcolt
    get coltamis = numero, fichcolt, all
    numcolhplc = (numero)
    diamcolhplc = (diametre)
    hautcolhplc = (hauteur)
    vitcolhplc = (vitesse)
    volmaxhplc = (volmax);

```

!-----introduction d'une nouvelle colonne-----

```

RULE 149
if colOK2 = AUTRE
then varcolOK2 = OK
    reset coltamis
    reset numero
    reset diametre
    reset hauteur
    reset vitesse
    reset volmax
    display "Veuillez dans ce cas introduire les differents parametres qui

```

vous sont proposes. Si vous ignorez une valeur, tapez '0'."

```
numero = (numrec)
display "Le numero de cette nouvelle colonne sera le {numero}"
find diametre
find hauteur
find vitesse
find volmax
append fichcolt
numcolhplc = (numero)
diamcolhplc = (diametre)
hautcolhplc = (hauteur)
vitcolhplc = (vitesse)
volmaxhpl = (volmax);
```

!----- que faire apres le TAMIS ? -----

RULE 150

if res[1] <= 20

then aprestamis = OK

display "- un TAMIS MOLECULAIRE etant donne que le volume est inferieur a 20 (ml).

Toutefois, du fait de son pouvoir separateur relativement faible, de sa lenteur et d'un aspect peu productif, nous vous conseillons d'appliquer cette technique dans les etapes finales de la purification, moment ou la proteine a separer est accompagnee d'un nombre limite de contaminants faiblement concentres."

else aprestamis = KO

display "- de concentrer votre echantillon etant donne que le volume est superieur

a 20 (ml), ceci dans le cas ou vous souhaiteriez appliquer un TAMIS MOLECULAIRE. De plus, nous vous indiquons que cette technique est surtout utilisee dans les etapes finales de la purification, moment ou la proteine a separer est accompagnee d'un nombre limite de contaminants faiblement concentres.";

!----- IMPLEMENTATION DE L'ELECTROPHORESE -----

RULE 151

if choixtech = electrophorese

then applitech = implemente

cls

display "Nous ne disposons pas de criteres suffisamment precis que pour vous guider dans la realisation de cette technique."

reset contapi

i = 1

whileknown vecpi[i]

reset vecpi[i]

i = (i+1)

end

reset maxpi

reset i

display "Nous allons cependant vous demander d'introduire les differentes valeurs

de PI que vous avez obtenues en resultat de l'application de la technique.

Si vous ne connaissez plus aucune valeur de PI, tapez '?'.

i = 1

find contapi

vecpi[i] = (contapi)

whileknown contapi

reset contapi

i = (i+1)

find contapi

vecpi[i] = (contapi)


```

end
maxpi = (i-1)
cls
reset calc2
reset i
reset tableau2
reset fourchette2
find calc2
reset choixtech
reset applitech
find applitech;

```

```

!----- procedure pour quitter le programme -----
RULE 152
if choixtech = quitter
then applitech = fini
  cls
  display "

```

NOUS VOUS REMERCIONS D'AVOIR UTILISE CE PROGRAMME.
 NOUS ESPERONS QU'IL A CONTRIBUE A VOUS AIDER DANS
 VOTRE TRAVAIL.

~";

```

!----- affichage recapitulatif des techniques realisees-----
RULE 153
if choixtech = recapitulatif
then applitech = aff_resume
  cls
  display "Des informations concernant les techniques realisees vont vous etr
e
presentees :

```

```

  "
  reset choixdate
  find choixdate
  display " TABLEAU RECAPITULATIF : "
  close fichres
  whileknown date
    get choixdate = date,fichres,all
    reset listerecap
    find listerecap
  end;

```

```

RULE 154
if choixdate = (date)
then listerecap = OK
  display "
DATE DU JOUR DEMANDE = {choixdate}
TECHNIQUE = {nomtech}
GEL = {gel}
VOLUME DU GEL = {volgel} (ml)
COLONNE = {colonne}
DIAMETRE DE LA COLONNE = {diamcol} (cm)
TAMPON = {tampon}
PH = {ph}
TYPE D'ELUTION = {typelut}

```

```

DUREE DE LA TECHNIQUE = {durtech} (heures)
PERTE INHERENTE A LA TECHNIQUE = {inhtech} (%)
VOLUME RESULTAT = {volres} (ml)
CONCENTRATION = {concent} (mg/ml)
PERTE TOTALE DUE A LA TECHNIQUE = {pertegl} (%)
~"

```

```

else listerecap = KO
  reset impression
  reset imprimer7
  find imprimer7;

```

!----- test de l'affichage du recapitulatif des techniques realisees-----

RULE 155

```

if impression = oui

```

```

then imprimer7 = OK

```

```

  display "Veuillez des lors vous assurer que la connection avec l'imprimante
est etablie et que cette derniere est 'ON LINE'."
~"

```

```

  pdisplay "Des informations concernant les techniques realisees vont vous et
re
presentees :

```

```

  "
  pdisplay " TABLEAU RECAPITULATIF : "
  close fichres
  whileknown date
    get choixdate = date, fichres, all
    reset listerecap2
    find listerecap2
  end
  reset choixtech
  reset applitech
  find applitech
else imprimer7 = KO
  reset choixtech
  reset applitech
  find applitech;

```

RULE 156

```

if choixdate = (date)

```

```

then listerecap2 = OK

```

```

  pdisplay "

```

```

DATE DU JOUR DEMANDE = {choixdate}

```

```

TECHNIQUE = {nomtech}

```

```

GEL = {gel}

```

```

VOLUME DU GEL = {volgel} (ml)

```

```

COLONNE = {colonne}

```

```

DIAMETRE DE LA COLONNE = {diamcol} (cm)

```

```

TAMPON = {tampou}

```

```

PH = {ph}

```

```

TYPE D'ELUTION = {typelut}

```

```

DUREE DE LA TECHNIQUE = {durtech} (heures)

```

```

PERTE INHERENTE A LA TECHNIQUE = {inhtech} (%)

```

```

VOLUME RESULTAT = {volres} (ml)

```

```

CONCENTRATION = {concent} (mg/ml)

```

```

PERTE TOTALE DUE A LA TECHNIQUE = {pertegl} (%)
~"

```

```

else listerecap2 = KO;

```

```

! *****
! ***** QUESTIONS *****
! *****

```

```

ask date_jour : "Veuillez introduire la date du jour sous le format suivant : JJ
MMAA.";

ask infos : "La proteine a purifier se trouve-t'elle dans la liste ci-dessus?";
choices infos : oui, non;

ask nom : "Quel est le nom de la proteine a purifier?";
ask proteine : "Dans ce cas , veuillez choisir le nom de la proteine parmi la li
ste proposee :";
ask nature : "Quelle est la nature de la proteine ?";
ask pi : "Quel est son point isoelectrique (PI) ?";
ask pm : "Quel est son poids moleculaire (PM) ?";
ask bi_stab : "Quelle est la borne inferieure de stabilite (bi_stab) ?";
ask bs_stab : "Quelle est la borne superieure de stabilite (bs_stab) ?";
ask bi_prec : "Quelle est la borne inferieure de precipitation (bi_prec) ?";
ask bs_prec : "Quelle est la borne superieure de precipitation (bs_prec) ?";
ask pour_stab : "Quel est le pourcentage de denaturation (pour_stab) ?";

ask infos2 : "Dans le cas ou des informations manqueraient dans la liste ci-dess
us,
en connaissez-vous les valeurs ?";
choices infos2 : oui,non;

ask choix : "Quelle caracteristique voulez-vous modifier ?
Si vous ne desirez (plus) rien modifier, selectionnez 'aucune'.";
choices choix : aucune,nature,PM,PI,bi_stab,bs_stab,bi_prec,bs_prec,pour_stab;

ask vol_ech : "Quel est le volume de l'echantillon (ml)?";
ask vol_res : "Quel est le volume de l'echantillon recueilli apres l'application
de la technique (en ml)?";

ask quant_ech : "Quelle est la quantite de l'echantillon (mg)?";
ask quant_res : "Quelle est la quantite de l'echantillon recueillie apres l'appli
cation
de la technique (en mg)?";

ask texture : "Quelle est la texture de votre melange ?";
choices texture : proteine_membranaire,proteine_soluble;

ask piconta : "Connaissez-vous le PI de TOUS les contaminants de la proteine a pu
rifier?";
choices piconta : oui,non;

ask pmconta : "Connaissez-vous le PM de TOUS les contaminants de la proteine a pu
rifier?";
choices pmconta : oui,non;

ask nbreconst : "Vous etes maintenant a meme de determiner le nombre de constitua
nts
du melange (proteine a purifier comprise). Veuillez l'indiquer ici : ";

ask contapi : "La valeur d'un PI est :";
ask contapm : "La valeur d'un PM est :";

ask OKvalpm2 : "La liste proposee vous semble-t-elle COMPLETE et DEFINITIVE ?";
choices OKvalpm2 : oui,non;

ask reaffich : "Voulez-vous encore modifier une caracteristique ?";
choices reaffich : oui, non;

ask colhplc : "Veuillez entrer le nom de la colonne pour l'HPLC :";
ask volgelhplc : "Quel est le volume du gel (en ml) ?";

```

ask voldephplc : "Quel volume comptez-vous déposer au sommet de la colonne (ml) ?";

ask quantdephplc : "Quelle quantité comptez-vous déposer au sommet de la colonne (ml) ?";

ask debithplc : "Afin de pouvoir déterminer avec plus de précision la durée de cette technique, veuillez entrer le débit de la colonne (ml/min) :";

ask choixchamp : "Lequel choisissez-vous ?";

ask choixtypelution : "Lequel choisissez-vous ?";

ask choixtech : "Veuillez effectuer votre choix parmi les techniques suivantes :
";

choices choixtech : tamis_moleculaire,chromato_ion,electrophorese,recapitulatif,quitter;

ask hautcolion : "Quelle est la hauteur de la colonne (entre 10 et 20 cm)?";

ask decrochage : "Quelle méthode d'élution désirez-vous ?";

choices decrochage : gradient,etapes;

ask fourchette : "Vous avez la possibilité de choisir la FOURCHETTE de Kav. Les valeurs + ou - 0.1 sont fréquemment utilisées. Si cela ne vous convient pas, faites votre choix parmi les valeurs suivantes :";

choices fourchette : 0.1,0.2,0.3,0.4,0.5;

ask autrefourch : "Souhaitez-vous étudier les PM limites calculées en fonction d'une autre valeur de fourchette de Kav ?";

choices autrefourch : oui,non;

ask fourchette2 : "Vous avez la possibilité de choisir le coefficient de résolution correspondant à votre gel.";

choices fourchette2 : 0.2,0.3,0.4,0.5,0.6,0.7,0.8;

ask autrefourch2 : "Souhaitez-vous étudier les pourcentages de contamination calculés sur base d'un autre coefficient de résolution de gel ?";

choices autrefourch2 : oui,non;

ask proportion : "Quelle est la proportion de la protéine dans le mélange (%) ?";

ask ajoutgel : "Voulez-vous introduire un nouveau type de gel ?";

choices ajoutgel : oui,non;

ask type : "Quel est son nom ?";

ask marque : "Quelle est sa marque ?";

ask domfract1 : "Quelle est sa borne inférieure de fractionnement ?";

ask domfracts : "Quelle est sa borne supérieure de fractionnement ?";

ask coeffns : "Quel est son coefficient de non-séparation ?";

ask ajoutcol : "Voulez-vous introduire une nouvelle colonne ?";

choices ajoutcol : oui,non;

ask diametre : "Quel est son diamètre ?";

ask hauteur : "Quelle est sa hauteur ?";

ask vitesse : "Quelle est sa vitesse d'écoulement ?";

ask volmax : "Quel est le volume maximal d'échantillon qui peut y être déposé?";

ask satisf : "Êtes-vous satisfait de cette proposition ?";

```
choices satisf : oui,non;
ask colOK2 : "Laquelle choisissez-vous (entrez le numero de la colonne)?
Si aucune ne vous convient, entrez 'AUTRE'.";
ask satisfgel2 : "Lequel choisissez-vous (Si vous voulez entrer un nouveau
gel, entrez 'AUTRE') ?";
choices satisfgel2 : Sephadex_G75,Sephacryl_S200,Sepharose 6B,
Sephadex G100,Ultrogel ACA-44,Ultrogel ACA-22,autre;

ask volgeltamis : "Quel est le volume du gel?";

ask quantdepon : "Quelle quantite comptez-vous déposer au sommet de la colonne
(mg) ?";

ask choixdate : "Veuillez entrer la date souhaitée dans le format suivant : JJMM
AA.";

ask impression : "Voulez-vous imprimer les données figurant ci-dessus ?";
choices impression : oui,non;
```